Homayoon Kazerooni
Thomas B. Sheridan

Computer Simulation and Control of Underwater Vehicles

MIT-T-82-007 C2

MIT Sea Grant
College Program

Massachusetts
Institute of Technology
Cambridge,
Massachusetts 02139

MITSG 82-19
October 1982

COMPUTER SIMULATION AND CONTROL OF UNDERWATER VEHICLES

Homayoon Kazerooni
Thomas B. Sheridan

Sea Grant College Program
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Report Number: MITSG 82-19
Index Number: NOAA/NA 81-AA-D-00065/R/U-4
October 1982

ABSTRACT

This report describes the digital simulation of underwater inspection vehicles. The dynamics and kinematics of the underwater vehicles have been studied in terms of the differential equations which can present the three dimensional motion of the vehicle (rotational and translational motion). In order to bring the motion of the vehicle under the control of the operator in a supervisory fashion, several control algorithms have been described. The digital dynamic models of two underwater vehicles ALVIN (Woods Hole Oceanographic Institution) and RCV150 (Hydro Products) have been used to simulate the control algorithms in performing some complicated tasks such as bottom-following. The results are shown in terms of computer graphic pictures.

ACKNOWLEDGMENTS

This work was supported by the MIT Sea Grant Program, National Oceanic and Atmospheric Administration, Department of Commerce and the Office of Naval Research.

AUTHORS

Homayoon Kazerooni completed this report as a Masters of Science Thesis in the Massachusetts Institute of Technology, Department of Mechanical Engineering. Mr. Kazerooni is currently pursuing underwater vehicle research as a doctoral candidate.

Thomas B. Sheridan, Professor Engineering and Applied Psychology at the Massachusetts Institute of Technology, was Mr. Kazerooni's Thesis Supervisor. He is the Principal Investigator on two Sea Grant sponsored projects, Remote Control Techniques for Unmanned Underwater Work Vehicles and Touch (Proximity and Contact Force) Sensing for Underwater Manipulators.

RELATED SEA GRANT REPORTS

TABLE OF CONTENTS

1 INTRODUCTION

1.1- UNDERWATER OPERATIONS

The world's ocean covers approximately 70 percent of the globe. As a promising resource supplier, the ocean is being explored. The number of underwater structures is being increased rapidly with the help of ocean technology.

Inspection of underwater and offshore structures is now performed by human divers, manned and unmanned submersibles. Usually most underwater operations include:

1) Exploration
2) Inspection
3) Construction
4) Maintenance
5) Rescue capability

As an undersea performer the diver's advantage is maneuverability, tactile sensing and cognition, but divers are losing their economic advantage because of the necessity of blood gas decompression and redundant back-up systems for their safety. Further ,the diver's working depth is at most 1000 feet even with the most advanced support systems. Undersea life support equipments for divers become increasingly expensive and personal safety becomes more and more difficult to maintain. Limited work time is another disadvantage associated with human divers. These problems and disadvantages suggest a variety of manned and unmanned submarines. Manned submarines are massive and expensive because of need for space for a pilot. The huge size of the manned underwater vehicle decreases the maneuverability of the vehicle. On the other hand the presence of a pilot or an observer in the vehicle has some advantages such as direct visual feedback.

Unmanned tethered vehicles are another group of undersea vehicles that are used in underwater inspection tasks. The restricted mobility caused by tether drag and tangle has not been solved yet. A recent development in undersea vehicles is the remotely manned untethered submarine for inspection tasks.

The data communication between an unmanned untethered submersible and an operator is conducted by a sonic link. The speed of sound in water is approximately 1600 meters per second, which means that an acoustic signal will take a delay time of one second for every 1600 meters of depth. The low bandwith of a sonic channel restricts transmittable information. By a sonic link information is transmitted approximately 30k bits per second under good conditions, much less under poor conditions. This means that a video picture composed of 128X128 points with 4 levels of gray scale would take 2 seconds for the entire "message" to be received and assembled on the surface.

1.2- MANUAL CONTROL OF UNDERWATER VEHICLES

In the past undersea vehicles were controlled by direct manual control. A human operator was always included in the closed-loop system as a controller. Human operators are not able to make decisions for different tasks in sufficient time, in other words the human operator has an inherent limitation (time delay) as a controller.

If the feedback message from a submersible is poor and/or the total control system has a significant time delay, control will deteriorate easily. An unmanned untethered inspection submersible communicating with an operator through a sonic link must be supported by an on-line computer. Usually two computers, one on the vehicle and one on the control station, handle all communications, display and control functions.

1.3- SUPERVISORY CONTROL OF UNDERWATER VEHICLES

In 1967 Ferrell and Sheridan proposed that a man-computer system based on the human supervisor-subordinate relationship could be used in deep space to solve some of the control problems involving time delay. Under supervisory control the human operator directs the subordinate computer by planning the action and directions it should take, teaching it how to achieve the desired functions, monitoring its performance, intervening whenever it gets into trouble, and trusting it to accomplish the tasks without continuous assistance.

Figure 1.1 shows the supervisory control concept applied to the system which has delay and a severely restricted data transmission rate. The remote computer is the actual direct controller of the vehicle and the camera. The local computer controls or monitors the remote computer. The information transmitted between them is modified and condensed as much as possible. The operator's control task is simplified by the assistance of the local computer so that the operator can concentrate on inspection tasks. Now the operator's role is converted from a master-slave controller to a "supervisor" of a semi-autonomous system.

When the operator chooses an automatic control mode, the remote computer takes care of the motion of the vehicle and the camera. When the automatic control mode is no more suitable (because of trouble or an irregular situation), the control mode is changed to the manual control mode automatically.

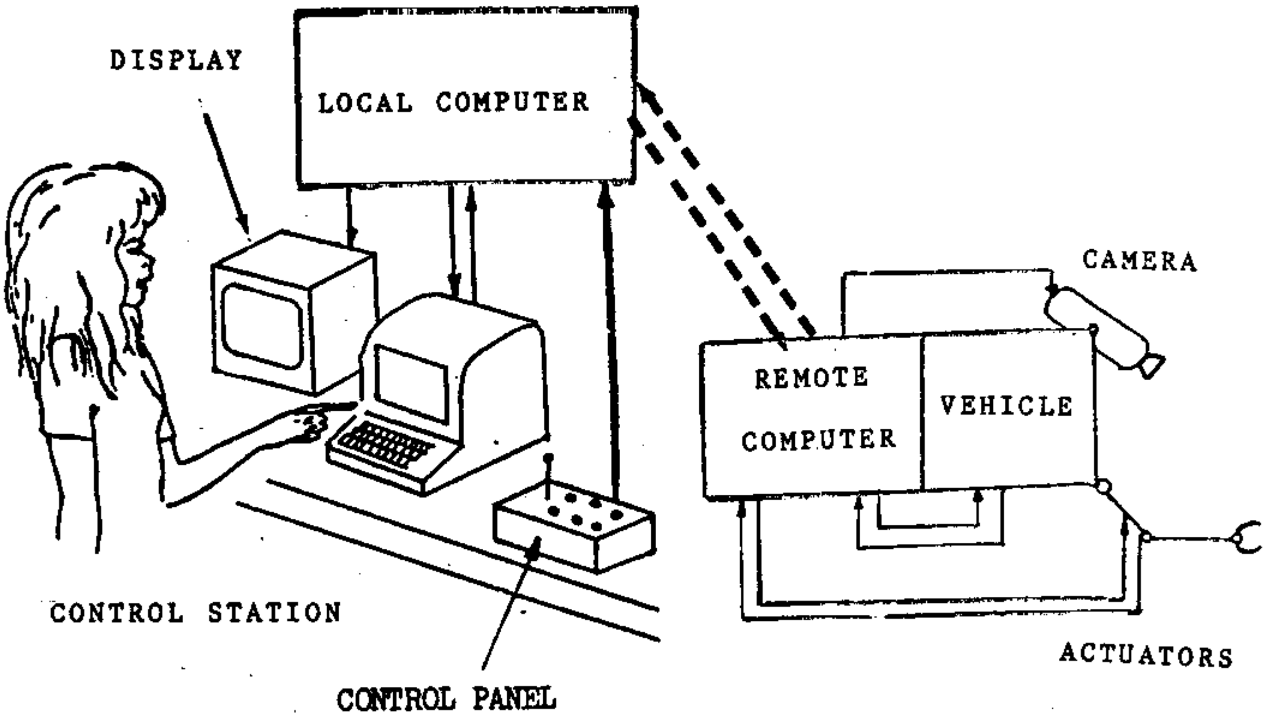


Fig. 1.1 Supervisory control of an underwater vehicle

2 OVERVIEW

The design of different control algorithms for an underwater vehicle requires an understanding of the dynamics of the vehicle and its maneuverability. In order to perform the following tasks in the lab , a combination of hardware and software simulation has been set up to analyze the maneuverability of the vehicle:

1) Understanding the dynamic behavior of the underwater vehicle

2) Analysis of the different model-based control algorithms for the vehicle

3) Design of some supervisory control algorithms to perform the complicated tasks such as following the bottom of the ocean

4) Analysis of sensitivity of the vehicle to different unwanted disturbances such as reaction of the arm on the vehicle and water turbulance

Figure 2.1 shows the simulation set upin theMan-Machine System Lab at MIT. An electric wheeled vehicle equipped with a T.V. camera is able to simulate the motion of the underwater vehicle in the laboratory. A PDP 11/34 computer takes care of all the software simulation. The digital simulation of the underwater vehicle runs the wheeled vehicle; therefore the camera on the vehicle moves the same way that the underwater vehicle moves under the water.

A vector display terminal has been employed to show a continuously moving three dimensional graphical simulation of the underwater vehicle. The graphical simulation is run with the dynamic simulation of the vehicle (the same software that runs the wheeled vehicle). The operator can communicate with the computer via a terminal, a joystick and some key knobs. All programs for dynamic simulation and controllers are run in real time.

In chapter three the dynamic behavior of the underwater vehicle has been analyzed. A general set of differential equations are derived to present the dynamics of the vehicles. By inserting different parameters into the program, one can simulate different underwater vehicles. Tight maneuverability of the vehicle and cross coupling between states of the equation are important subjects which are discussed in chapter three.

In chapter four a design procedure for the orientation controller is studied. The orientation controller allows the operator to orient the vehicle in any desired orientation, as long as the vehicle is mechanically able to perform the desired task. Three angles (pitch, roll and yaw)are enough to specify the orientation of the vehicle.

Chapter five devotes itself to design of the position controller. With the help of the position controller the operator is able to position the vehicle relative to a stationary point. The position of the vehicle in three dimensional space can be specified with the help of three measurements. The operator assigns the coordinate of the desired point (X,Y and Z) relative to his/her position (control station) using the position controller.

A bottom following program is described in chapter six. Bottom following is an example of a complicated task which is performed by combining different controllers. This algorithm allows the vehicle to follow a path in a vertical plane within a specified margin close to the bottom of the ocean.

In chapter seven a special purpose autopilot is designed for a manned submarine (ALVIN). The ALVIN's autopilot is able to follow different paths underwater at a constant range. In this algorithm the vehicle senses the environment by sonars. Two sonars on the vehicle are able to fetch the necessary data for the bottom following algorithm. The algorithm is able to construct an approximate model of the environment for the purpose of bottom following. Having the model of the path designed by the software, the thrusters of the vehicle are driven according to the decision made by the algorithm and implemented via the servocontrollers.

Chapter eight contains all necessary data to simulate RCV150 and ALVIN.

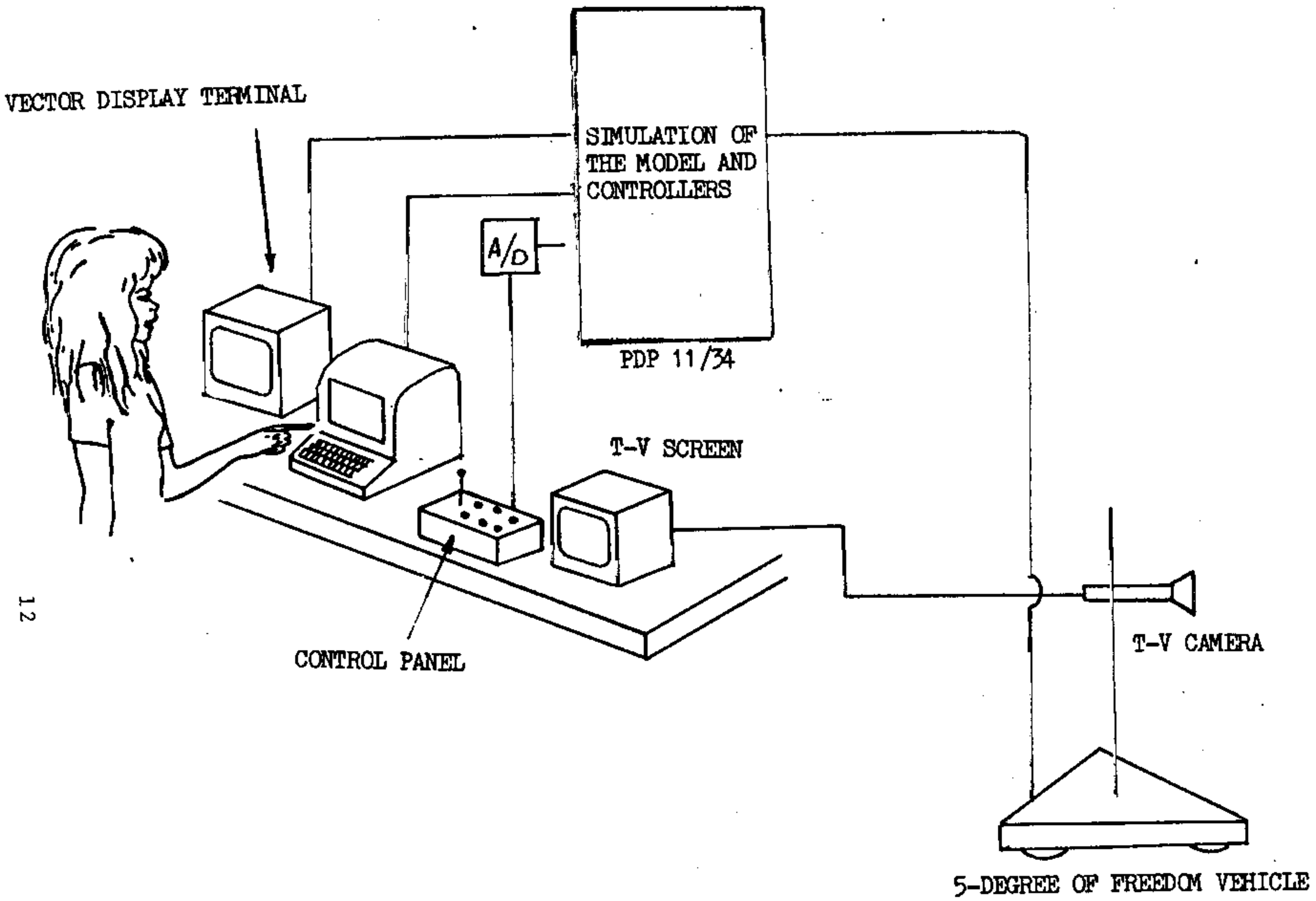


Fig. 2.1 Simulation set up in Man-Machine System Lab. at MIT

3 MODELING

3.1- INTRODUCTION

This chapter addresses itself to modelling of an underwater vehicle in terms of the differential equations which simulate the entire dynamic behavior of the vehicle. The model can be used not only for analysis of the dynamic behavior but also for controller design. As will be explained later the model has a high degree of coupling between different motions of the vehicle.

The model is applicable for most underwater vehicles. By assigning different parameters to the model, one can simulate different underwater vehicles on a digital computer. The simulation has been implemented in two different modes:

1) Digital simulation on VAX 11/782 for analysis of the dynamic behavior of the vehicle; and

2) Digital real time simulation on PDP 11/34 .

3.2- COORDINATE SYSTEMS

An underwater vehicle is capable of moving with six degrees of freedom of motion: translation along three orthogonal axes and rotation around each of the three axes. Most underwater vehicles have at least one plane of symmetry. Port and starboard have the same geometry and represent reflections of each other in the centerline plane. This symmetry in body shape can be observed in ships and submarines. A coordinate system which takes advantage of the plane of symmetry was chosen.

Our mathematical model uses two orthogonal coordinate systems: one remains fixed at the water surface while the other travels with the vehicle to act as a local reference system. The origin of the moving frame is located at the center of mass of the vehicle. Figure 3.1 defines the axis system chosen; ox and oy are fixed in the plane of symmetry. The axes oriented by symmetry are usually parallel to the principal axes of inertia.

The x-axis is the longitudinal axis in the plane of symmetry with positive forward and is the intersection of the two planes of symmetry.

The y-axis is perpendicular to the plane of symmetry with positive to starboard.

The z-axis is in the plane of symmetry, perpendicular to the xy plane.

Then OXYZ is a fixed coordinate system with origin at the surface of the sea.

X and Y are mutually perpendicular axes in the horizontal surface of the sea.

Z is an axis perpendicular to the surface with positive direction downward.

Using the axes shown in figure 3.1 , one can specify $\vec{Rc}$ by quantities Xc,Yc,Zc :

$$\vec{Rc} = Xc.\hat{I} + Yc.\hat{J} + Zc.\hat{K}$$

$\vec{U}=u.\hat{i}+v.\hat{j}+w.\hat{k}$ is the vector linear velocity of the body, where u,v,w are the components of the linear velocity along the x,y,z axes in the body, respectively.

$\vec{r}=x.\hat{i}+y.\hat{j}+z.\hat{k}$ is the location of a point within the vehicle .

$\vec{\Omega} =p.\hat{i}+q.\hat{j}+r.\hat{k}$ is the vector angular velocity of the body, where p,q,r are the components of the angular velocity of the body around x,y,z axes respectively.

The equations of motion for the rigid vehicle contain the total external force F and the moment about C of the external forces M. In terms of the unit vectors associated with the body axes, these can be written in the form:

$$\vec{F}=X.\hat{i}+Y.\hat{j}+Z.\hat{k}$$

$$\vec{M}=Mx.\hat{i}+My.\hat{j}+Mz.\hat{k}$$

The vectors U,r,F,G have been specified in terms of $\hat{i},\hat{j},\hat{k}$, although they could also be specified in terms of

$\hat{I},\hat{J},\hat{K}$. It is preferred to define all forces on the vehicle with respect to the moving coordinate system ,because in practice all forces on the body such as thrusters and drags can be calculated more easily with respect tothe body coordinate system than with respect to the fixed coordinate system on the surface.

3.3- POSITION AND ORIENTATION OF THE VEHICLE

The position of the vehicle can only be specified by reference to the axes OXYZ, and using the rectangular coordinates Xc,Yc,Zc. The orientation of the vehicle can also be specified with respect to OXYZ, but the method is less obvious. The usual approach is to start with Cxyz parallel to OXYZ and bring the vehicle from this reference orientation to its actual one by:

1) a 'swing' ψ around z axis

2) a 'tilt ' θ around y axis

3) a 'heel ' ϕ around x axis

It is important to note that these rotations must be performed in this order. The quantities ϕ , θ , ψ can now be used as the required " orientation coordinates ." When used in this manner, they are a modified form of "Euler's Angles ."

In summation , then, the underwater vehicle may be located in space by:

a) three position coordinates Xc,Yc,Zc

b) three angular coordinates ϕ , θ , ψ

But the latter set is only meaningful if the rotation in swing,tilt and heel is applied in the specified order.

3.4- AXIS TRANSFORMATION

It has been pointed out that the order in which the rotations are applied matters because it determines the resulting orientation. It can also be demonstrated easily that the orientation may not be specified in terms of vectors.

Consider, for example, the special case in which ψ, θ, ϕ are all equal to π radians, and, instead of an underwater vehicle, we rotate a book as in figure 3.2 . If we were (incorrectly) to use a vector formulation of this process, we should arrive at the unwanted vector polygon shown in figure 3.2 . Since the polygon must be closed, there is apparently a resultant rotation, whereas figure 3.2 shows that this is not the case.

If the velocity components u,v,w are given, the coordinates Xc,Yc,Zc vary in a manner that depends on the prevailing values of Φ,Θ,Ψ. The purpose now is to examine this dependence, keeping in mind that ϕ,θ,ψ are not to be thought of as components of a vector.

The velocity of C may be written as :

$$\vec{U}=U1.\hat{i}1+V1.\hat{J}1+W1.\hat{k}1$$

with respect to the axes in their initial orientation as shown by dotted lines in figure 3.3a . Alternatively the velocity of C may be written:

$$\vec{U}=U2.\hat{i2}+V2.\hat{j2}+W2.\hat{k2}$$

by reference to cx2y2z2. By inspection it is shown

$$\begin{pmatrix} U1 \\ V1 \\ W1 \end{pmatrix} = \begin{pmatrix} COS(\psi) & -SIN(\psi) & 0 \\ SIN(\psi) & COS(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} U2 \\ V2 \\ W2 \end{pmatrix}$$

or

$$U1=T1(\Psi)*U2$$

Next, the tilt angle Θ must be applied by rotating Cx2y2z2 to Cx3y3z3 as shown in the figure 3.3b . The velocity of C may be written:

$$\vec{U}=U3.\hat{i3}+V3.\hat{j3}+W3.\hat{k3}$$

It can be seen that:

$$\begin{pmatrix} U2 \\ V2 \\ W3 \end{pmatrix} = \begin{pmatrix} COS(\Theta) & 0 & SIN(\Theta) \\ 0 & 1 & 0 \\ -SIN(\Theta) & 0 & COS(\Theta) \end{pmatrix} \times \begin{pmatrix} U3 \\ V3 \\ W3 \end{pmatrix}$$

or $U2=T2(\Theta)U3$

Finally the heel Φ must be applied , as illustrated in figure 3.3c , so that Cx3y3z3 takes up the actual position Cxyz. The velocity of C is

$$\vec{U}=u\hat{i}+v\hat{j}+w\hat{k}$$

Now by inspection it is seen that

$$\begin{pmatrix} U3 \\ V3 \\ W3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{pmatrix} \times \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

$$U3=T3(\phi)U$$

The vector U1 is the same as

$$\dot{\vec{R}}c=\dot{X}c.\hat{I}+\dot{Y}c.\hat{J}+\dot{Z}c.\hat{K}$$

so that

$$\vec{U}1=\dot{\vec{R}}c$$

and it follows that

$$\begin{pmatrix} \dot{X}c \\ \dot{Y}c \\ \dot{Z}c \end{pmatrix} =T1(\psi)U2=T1(\psi)T2(\theta)U3=T1(\psi)T2(\theta)T3(\phi)U=TU$$

where

$$T=\left(\begin{array}{c|c|c} \cos(\psi)\cos(\theta) & \cos(\psi)\sin(\theta)\sin(\phi) & \cos(\psi)\sin(\theta)\cos(\phi) \\ & -\sin(\psi)\cos(\phi) & +\sin(\psi)\sin(\phi) \\ \hline \sin(\psi)\cos(\theta) & \sin(\psi)\sin(\theta)\sin(\phi) & \sin(\psi)\sin(\theta)\cos(\phi) \\ & +\cos(\psi)\cos(\phi) & -\cos(\psi)\sin(\phi) \\ \hline -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{array}\right)$$

A knowledge of ϕ , θ , ψ ,u,v,w as functions of time will permit step by step integration that results in Xc,Yc,Zc and hence $\vec{R}c$ as a function of time.

3.5- ANGULAR VELOCITY EXPRESSED IN TERMS OF MODIFIED EULER ANGLES

Even though finite rotations are not vector quantities, angular velocity is. If the angular velocity of a vehicle is

$\overrightarrow{\Omega} = p.\hat{i} + q.\hat{j} + r.\hat{k}$

then the way the components p,q,r vary with time depends on the way in which ϕ , θ , ψ vary with time. It is possible to derive the relationship that exists between the two sets of quantities p , q , r and ϕ , θ , ψ.

By superimposing a suitable translational velocity on a rigid body, any chosen point of the body may be brought to rest without changing its angular velocity. It is convenient to imagine point C fixed in this way, so that only the oriention of the frame Cxyz varies with time. This means that, as time goes on, the frames Cx2y2z2, Cx3y3z3,Cxyz all rotate about C (while, of course, Cx1y1z1 remains stationary and parallel to OXYZ). To specify the orientation Cxyz at any instant it is necessary to freeze the frames and measure the angles ϕ , θ , ψ.

Since angular velocity is a vector quantity,we can add:
angular velocity of Cxyz relative to Cx3y3z3 ($=\dot{\phi}$i);
angular velocity of Cx3y3z3 relative to Cx2y2z2 ($=\dot{\theta}$j3);
angular velocity of Cx2y2z2 relative to Cx1y1z1 ($=\dot{\psi}$k2);

where Cx1y1z1 is fixed. Therefore

$$\vec{\Omega} = \dot{\phi}.\hat{i}+\dot{\theta}.\hat{j3}+\dot{\psi}.\hat{k2}$$

and in order to find expressions for p,q,r, we must express $\hat{j3}$ and $\hat{k2}$ in terms of $\hat{i},\hat{j},\hat{k}$.

If $\hat{j3}=a.\hat{i}+b.\hat{j}+c.\hat{k}$

then:

$$\begin{pmatrix}0\\1\\0\end{pmatrix}=T3(\phi)\begin{pmatrix}a\\b\\c\end{pmatrix}$$

This is inverted because $|T3|=1$, therefore

$$\begin{pmatrix}a\\b\\c\end{pmatrix}=\begin{pmatrix}0\\COS(\phi)\\-SIN(\phi)\end{pmatrix}$$

Again, if

$$\hat{k2}=d.\hat{i}+e.\hat{j}+f.\hat{k}$$

then:

$$\begin{pmatrix}0\\0\\1\end{pmatrix}=T2(\theta)*T3(\phi)\begin{pmatrix}d\\e\\f\end{pmatrix}$$

Since $|T2|=1$ this is easily inverted to give:

$$\begin{pmatrix}d\\e\\f\end{pmatrix}=\begin{pmatrix}-SIN(\theta)\\SIN(\phi)*COS(\theta)\\COS(\phi)*COS(\theta)\end{pmatrix}$$

To summarize,

$$=(\dot{\phi}-\dot{\psi}SIN(\theta))\hat{i}+(\dot{\theta}COS(\phi)+\dot{\psi}SIN(\phi)COS(\theta))\hat{j}+(\dot{\psi}\,COS(\phi)$$

$\cos(\theta)-\dot{\theta}\sin(\phi))\hat{k}$

$$=p.\hat{i}+q.\hat{j}+r.\hat{k}$$

The components p , q , r are thus expressed in terms of ψ, θ ,ϕ and their derivatives. The result may be put in matrix form and then inverted:

$$\begin{pmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{pmatrix}=\begin{pmatrix}1 & \mathrm{SIN}(\phi)*\mathrm{TAN}(\theta) & \mathrm{COS}(\phi)*\mathrm{TAN}(\theta)\\ 0 & \mathrm{COS}(\phi) & -\mathrm{SIN}(\phi)\\ 0 & \mathrm{SIN}(\phi)*\mathrm{SEC}(\theta) & \mathrm{COS}(\phi)*\mathrm{SEC}(\theta)\end{pmatrix}\times\begin{pmatrix}p\\ q\\ r\end{pmatrix}$$

With the help of the matrix above, one can trace the variation of the orientation coordinates in terms of the angular velocities of

roll, p

pitch, q

yaw, r

3.6- EQUATIONS OF MOTION

The velocity of the vehicle can be written:

$\vec{U}=u.\hat{i}+v.\hat{j}+w.\hat{k}$

In this part we would like to develop equations of motion in vector form. It is also desirable to separate the gravity force from the fluid forces. The force equation is

$$\vec{F}=M\frac{d\vec{U}}{dt}$$

The symbols X,Y,Z are reserved for the components of fluid forces, illustrated as :

$$\vec{F}=X.\hat{i}+Y.\hat{j}+Z.\hat{k}+mg.\hat{K},$$

where $\hat{K}$ is the unit vector pointing vertically downwards. Since it will be convenient to express $\vec{F}$ entirely in terms of unit vectors $\hat{i},\hat{j},\hat{k}$, then:

$$mg.\hat{K}=Fx.\hat{i}+Fy.\hat{j}+Fz.\hat{k}$$

To find Fx, Fy, and Fz we will trace the effects of applying successively the orientation angles ϕ , θ , ψ to bring the vehicle to its actual orientation. First, apply the swing ψ, so that Cx1y1z1 swings to Cx2y2z2. The components (Fx)2, (Fy)2, (Fz)2 in the new directions are obviously 0,0,mg respectively. These results may be found by using appropriate scalar products. Thus, from figure 3.4a

$$(Fx)2=mg.\hat{k}2.\hat{i}2=0$$

$$(Fy)2=mg.\hat{k}2.\hat{j}2=0$$

$$(Fz)2=mg.\hat{k}2.\hat{k}2=mg$$

Next we apply the angle of tilt as figure 3.4b shows :

$$(Fx)3=mg.\hat{k}2.\hat{i}3$$

$$=mg.\hat{k}2.(COS(\theta)\hat{i}2-SIN(\theta)\hat{k}2)$$

$$=-mg.SIN(\theta)$$

$$(Fy)3=mg.\hat{k}2.\hat{j}3$$

$$=mg.\hat{k}2.\hat{j}2$$

$$=0$$

$$(Fz)3=mg.\hat{k}2.\hat{k}3$$

$$=mg.\hat{k}2.(SIN(\theta)\hat{i}2+COS(\theta)\hat{k}2)$$

$=mg.COS(\theta)$

Finally, we apply the angle of heel to bring the vehicle to its actual orientation (figure 3.4 c)

$Fx=((Fx)3\hat{i3}+(Fz)3\hat{k3}).\hat{i}$

$=(-mg\ SIN(\theta)\hat{i3}+mg\ COS(\theta)\hat{k3}).\hat{i3}$

$=-mg\ SIN(\theta)$

$Fy=(-mg.COS(\theta)\hat{i3}+mg.COS(\theta)\hat{k3}).\hat{j}$

$=(-mg.SIN(\theta)\hat{i3}+mg.COS(\theta)\hat{k3}).(COS(\phi)\hat{j3}+SIN(\phi)\hat{k3})$

$=mg.COS(\theta).COS(\phi)$

$Fz=(-mg.SIN(\theta)\hat{i3}+mg.COS(\theta)k3).\hat{k}$

$=(-mg.SIN(\theta)\hat{i3}+mg.COS(\theta)\hat{k3}).(-SIN(\phi).\hat{j3}+COS(\phi)\hat{k3})$

$=mg.COS(\theta).COS(\phi)$

Then:

$F=(X-mg.SIN(\theta))\hat{i}+(Y+mg.COS(\theta).SIN\ (\phi))\hat{j}+(Z+mg.COS(\theta).COS(\phi))\hat{k}$

and in terms of unit vectors:

$$\vec{F}=m.(U.\hat{i}+V.\hat{j}+W.\hat{k})+m.\begin{pmatrix}\hat{i} & \hat{j} & \hat{k}\\ p & q & r\\ u & v & w\end{pmatrix}$$

The scalar equation can now be written :

$X-mg\ SIN(\theta) \qquad =m(\dot{u}+qw-rv)$

$Y+mg\ COS(\theta)\ SIN(\phi) \qquad =m(\dot{v}+ru-pw)$

$Z+mg\ COS(\theta)\ COS(\phi) \qquad =m(\dot{w}-pv-qu)$

Unlike the force vector $\vec{F}$, $\vec{M}$ contains no contribution from the gravity force because it represents a moment about the center of gravity. To obtain the scalar equations from the vector equation is straightforward. The vector equation is

$$\vec{M} = Mx.\hat{i}+My.\hat{j}+Mz.\hat{k}$$

The angular momentum about a set of axes fixed in the vehicle can be expressed as:

$$\text{angular momentum} = \begin{pmatrix} Ix & -Ixy & -Ixz \\ -Iyx & Iy & -Iyz \\ -Izx & -Izy & Iz \end{pmatrix} \times \begin{pmatrix} p \\ q \\ r \end{pmatrix}$$

where the terms Iab,(a≠b) are the products of inertia. If the axes chosen are principal axes of inertia, then the products of inertia are zero and we have:

$$(\text{angular momentum}) = \begin{pmatrix} Ix & 0 & 0 \\ 0 & Iy & 0 \\ 0 & 0 & Iz \end{pmatrix} \times \begin{pmatrix} p \\ q \\ r \end{pmatrix}$$

or

$$(\text{angular momentum}) = \hat{i}.Ix.p+\hat{j}.Iy.q+\hat{k}.Iz.r$$

$$\begin{aligned} \vec{M} &= \frac{d}{dt}(\text{Ang. mom.}) = \frac{d}{dt}(\hat{i}.Ix.p+\hat{j}.Iy.q+\hat{k}.Iz.r) \\ &= \hat{i}.\frac{d}{dt}(Ix.p)+Ix.p.\frac{d\hat{i}}{dt}+\hat{j}.\frac{d}{dt}(Iy.q)+Iy.q.\frac{d\hat{j}}{dt}+\hat{k}.\frac{d}{dt}(Iz.r)+Iz.r.\frac{d\hat{k}}{dt} \end{aligned}$$

Since the mass of the vehicle is assumed constant in time, then the inertia of the vehicle is also assumed constant in time. Hence, $\frac{d}{dt}(Ix.p)=Ix.\dot{p}$ with analogous results for the similar terms:

$$\vec{M}=\hat{i}Ix.\dot{p}+Ixp(\hat{j}r-\hat{k}q)+\hat{j}Iy.\dot{q}+Iyq(\hat{k}p-\hat{i}r)+\hat{k}Iz.\dot{r}+Izr(\hat{i}q-\hat{j}p)$$

With the vector components of moment defined as

$$\vec{M}=\hat{i}.Mx+\hat{j}.My+\hat{k}.Mz$$

the grouping of the vector quantities into components in the $\hat{i},\hat{j},\hat{k}$ directions gives:

$$Mx=Ix.\dot{p}+(Iz-Iy)qr$$

$$My=Iy.\dot{q}+(Ix-Iz)rp$$

$$Mz=Iz.\dot{r}+(Iy-Ix)pq$$

Terms $(Iz-Iy)qr$, $(Ix-Iz)rp$ and $(Iy-Ix)pq$ represent gyroscopic moments arising from the moving axis system.

The equations which have been developed are useful for the case where the center of gravity is located at the center of mass. Since hydrodynamic forces depend greatly on the geometry or buoyancy of the body, it is very useful to develop the equations for an arbitrary origin within the body of the vehicle. The vector distance from the center of gravity to the origin is $Rg=Xg.\hat{i}+Yg.\hat{j}+Zg.\hat{k}$, where Xg, Yg and Zg are the distance components along the x, y and z axes respectively (see figure 3.5). On page 31 the equations describing the dynamics of the underwater vehicle have been written.

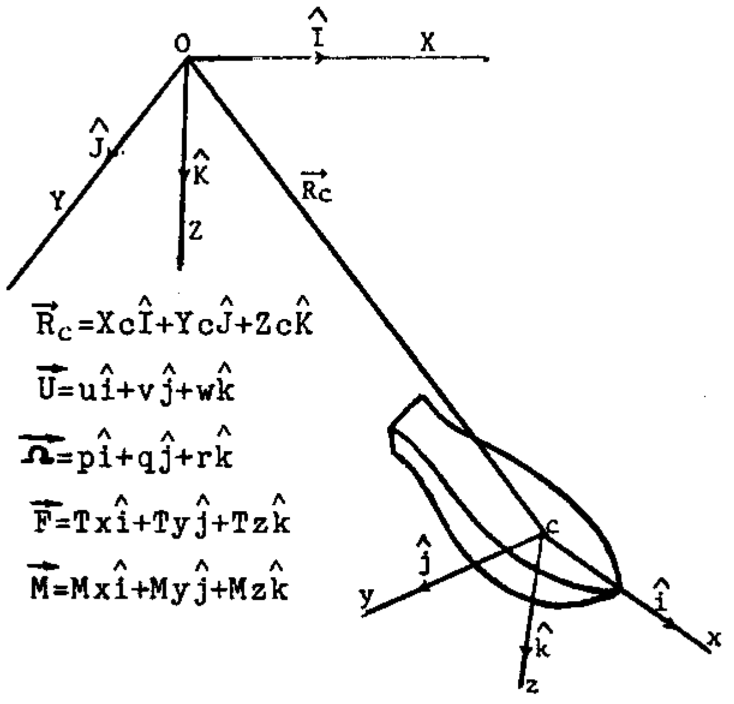

O
$\hat{I}$
X
$\hat{J}$
Y
$\hat{K}$
Z
$\vec{R}_c$
$\vec{R}_c = X_c\hat{I} + Y_c\hat{J} + Z_c\hat{K}$
$\vec{U} = u\hat{i} + v\hat{j} + w\hat{k}$
$\vec{\Omega} = p\hat{i} + q\hat{j} + r\hat{k}$
$\vec{F} = Tx\hat{i} + Ty\hat{j} + Tz\hat{k}$
$\vec{M} = Mx\hat{i} + My\hat{j} + Mz\hat{k}$
c
$\hat{j}$
y
$\hat{i}$
x
$\hat{k}$
z

Fig. 3.1 Two coordinate systems are used to model the dynamics of underwater vehicles.

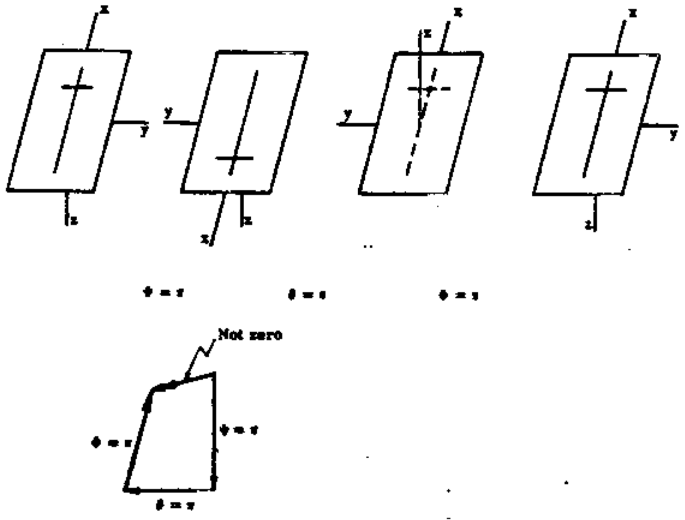


Fig. 3.2

Fig. 3.2

Adding the three angles ψ , θ and ϕ in vectorial form results in an incorrect angle.

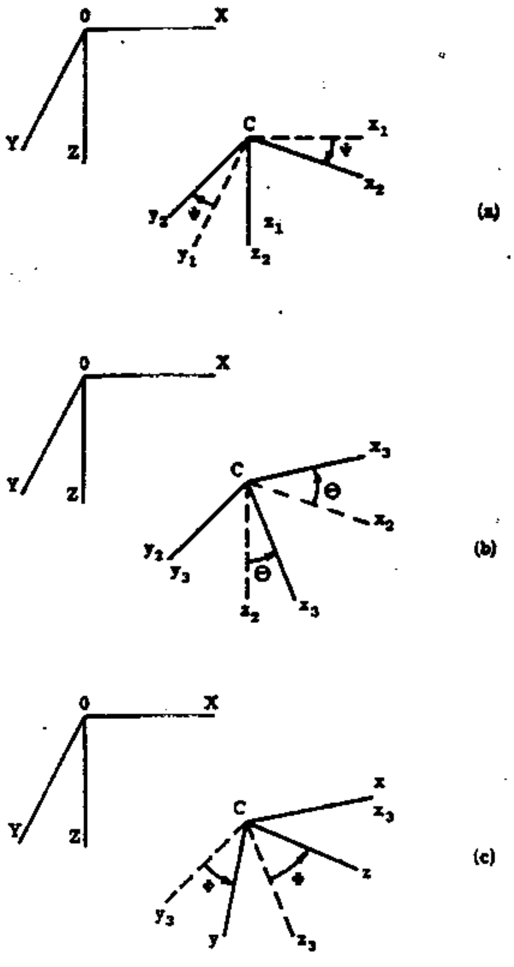


Fig. 3.3

The order of rotation in this report is ψ , θ and then ϕ .

Fig. 3.3

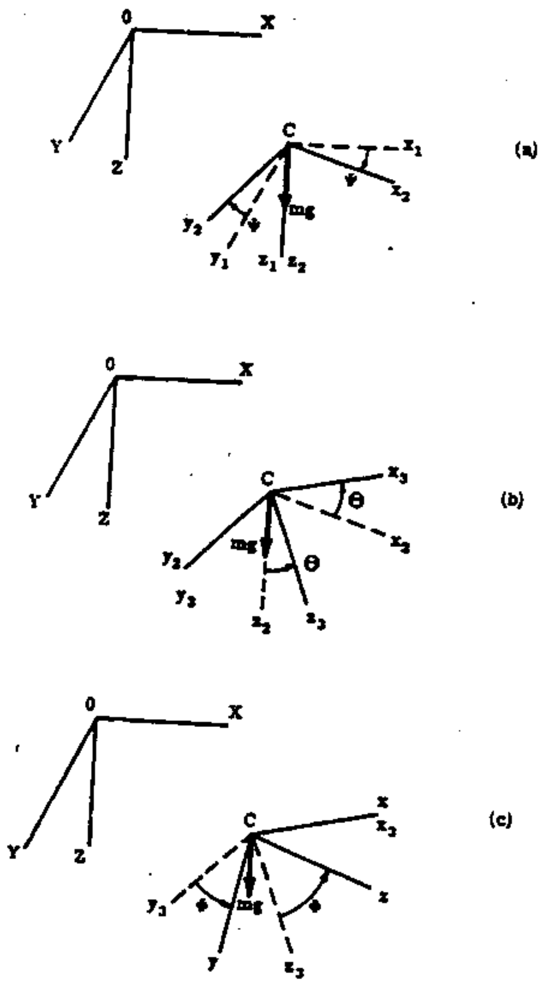


Fig. 3.4 Effect of vehicle weight in rotational motion.

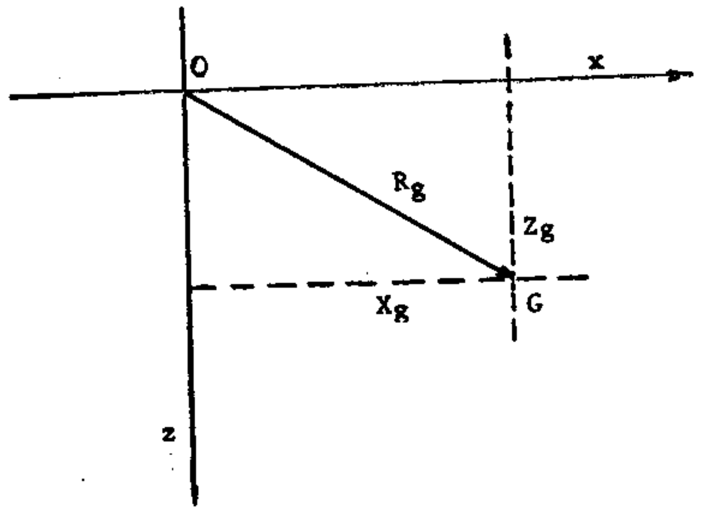


Fig. 3.5 $\overrightarrow{Rg}$ is the vector that shows the distance between center of geometry and center of mass.

Summary of equations

1) $Tx-DRx-Wgt*SIN(\theta) = M*[\dot{u}+q*w-r*v-$
$Xg*(q**2+r**2)+Yg*(p*q-\dot{r})+Zg*(p*r+\dot{q})]$

2) $Ty-DRy-Wgt*COS(\theta)*SIN(\phi)=M*[\dot{v}+r*u-p*w-$
$Yg*(r**2+p**2)+Zg*(q*r-\dot{p})+Xg*(p*q+\dot{r})]$

3) $Tz-DRz-Wgt*COS(\theta)*COS(\phi)=M*[\dot{w}+p*v-u*q-$
$Zg*(p**2+q**2)+Xg*(r*p-\dot{q})+Yg*(q*r+\dot{p})]$

4) $Mx-DRxx-Wgt*SIN(\theta)*Zg=Ix*\dot{p}+(Iz-Iy)*q*r+M*[Yg*(\dot{w}+p*v-u*q)$
$-Zg*(\dot{v}+u*r-w*p)]$

5) $My-DRyy+Wgt*COS(\theta)*SIN(\phi)*Zg=Iy*\dot{q}+(Ix-Iz)*r*p+M*[Zg*(\dot{u}+q*w-r*v)$
$-Xg*(\dot{w}+p*v-u*q)]$

6) $Mz-DRzz=Iz*\dot{r}+(Iy-Ix)*p*q+M*[Xg*(\dot{v}+r*u-p*w)$
$-Yg*(\dot{u}+q*w-r*v)]$

Tx=Thrust force on the body of the vehicle in x-direction
Mx=Thrust moment on the body of the vehicle around x-direction
Wgt=Weight of the vehicle
DRx=Drag force in x-direction
DRxx=Drag moment around x-direction
M=Mass of the vehicle
u=Linear velocity in x-direction
v=Linear velocity in y-direction
w=Linear velocity in z-direction
p=Angular velocity around x
q=Angular velocity around y
r=Angular velocity around z

Transformation Matrices

$$
\begin{array}{l}
1\\ \\ 2\\ \\ 3
\end{array}
\begin{bmatrix} \dot{X}c \\ \\ \dot{Y}c \\ \\ \dot{Z}c \end{bmatrix} =
\begin{bmatrix}
COS(\psi)*COS(\theta) & COS(\psi)*SIN(\theta)*SIN(\phi) & COS(\psi)*SIN(\theta)*COS(\phi) \\
 & -SIN(\psi)*COS(\phi) & +SIN(\psi)*SIN(\phi) \\
SIN(\psi)*COS(\theta) & SIN(\psi)*SIN(\theta)*SIN(\phi) & SIN(\psi)*SIN(\theta)*COS(\phi) \\
 & +COS(\psi)*COS(\phi) & -COS(\psi)*SIN(\phi) \\
-SIN(\theta) & COS(\theta)*SIN(\phi) & COS(\theta)*COS(\phi)
\end{bmatrix}
\times
\begin{bmatrix} u \\ \\ v \\ \\ w \end{bmatrix}
$$

$$
\begin{array}{l}
4\\ 5\\ 6
\end{array}
\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} =
\begin{bmatrix}
1 & SIN(\phi)*TAN(\theta) & COS(\phi)*TAN(\theta) \\
0 & COS(\phi) & -SIN(\phi) \\
0 & SIN(\phi)*SEC(\theta) & COS(\phi)*SEC(\theta)
\end{bmatrix}
\times
\begin{bmatrix} p \\ q \\ r \end{bmatrix}
$$

4 ORIENTATION CONTROLLER

4.1- INTRODUCTION

As explained in Chapter 3 the equations of motion of the underwater vehicle have non-linear terms. Here we discuss the type of non-linearities that exist. Study of the non-linearities in the model of the vehicle helps us to understand the dynamic behavior of the underwater vehicle and consequently the design of the controllers. After some analysis on special characteristics of the vehicle, the reader will appreciate the sluggish dynamic behavior of the vehicle under tight maneuvering. Finally we will offer a design procedure which will enable us to have more control on the underwater vehicle orientation.

As mentioned in the previous chapter the orientation of the underwater vehicle in three dimensional space can be specified with just three variables, ϕ , θ and ψ . Most underwater vehicles used for undersea inspection are equipped with thrusters aimed in different directions. Up to a point the more thrusters an underwater vehicle has, the more maneuvering it is able to do. For example some remotely-manned underwater vehicles are equipped with several thrusters which produce fine adjustments in orientation. In contrast, a large Navy submarine, equipped only with a rear propeller and utilizing surface controls for rotation, is much less responsive to adjustments in orientation. This is an important dynamic behavior difference between an underwater inspection vehicle and a standard Navy submarine.

4.2- CROSS-COUPLING

The equations on page 31 show that the orientation variables(ϕ, θ, ψ) are cross-coupled. For example, any torque around the x-axis not only produces rotation around the x-axis but also affects rotation around the y and z-axes. However the strength of the coupling in the model depends on the orientation and physical configuration of the vehicle. In some orientations there is almost zero coupling in the model.

The same coupling exists when torque is supplied around the other axes. For example, any torque around the y-axis not only produces rotation around the y-axis but also might affect rotation around the x and z-axes. The rotational equations which are explained in the previous chapter show the coupling between rotational state variables p,q,r,ϕ,θ,ψ. In general the part of the model which demonstrates the rotation of the vehicle can be shown:

$$\begin{tikzcd}
Mx \arrow[rd] & & \phi = f1(Mx,My,Mz,\phi,\theta,\psi,\ldots) \\
My \arrow[r] & \boxed{\text{non-linear model}} \arrow[ru] \arrow[r] \arrow[rd] & \theta = f2(Mx,My,Mz,\phi,\theta,\psi,\ldots) \\
Mz \arrow[ru] & & \psi = f3(Mx,My,Mz,\phi,\theta,\psi,\ldots)
\end{tikzcd}$$

where:

ϕ=rotation around x-axis (rolling)

θ=rotation around y-axis (pitching)

ψ=rotation around z-axis (yawing)

$\vec{M}=Mx.\hat{i}+My.\hat{j}+Mz.\hat{k}$ is the moment acting on the body of the vehicle. This can be from the thrusters or reaction of the manipulators on the body or from water flow. The angular rotations calculated from the model show that the cross-couplings in most cases are not ignorable and depend not only on the physical and geometric characteristics of the vehicle but also on its orientation.

The existence of cross-coupling in the model is an important factor in selection of the orientation controllers. We shall go through some details of the cross-coupling phenomena in the model:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \mathrm{SIN}(\phi)*\mathrm{TAN}(\theta) & \mathrm{COS}(\phi)*\mathrm{TAN}(\theta) \\ 0 & \mathrm{COS}(\phi) & -\mathrm{SIN}(\phi) \\ 0 & \mathrm{COS}(\phi)/\mathrm{SIN}(\theta) & \mathrm{COS}(\phi)/\mathrm{COS}(\theta) \end{bmatrix} \times \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

p,q and r are rotational velocities produced by different torques around the moving axes. The angles which identify the orientation of the underwater vehicle are ϕ,θ,ψ. The matrix above shows the contribution of different angular velocities on the orientation of the vehicle. One may notice that the rolling angle of the underwater vehicle is a function of p,q,r,θ,ϕ. Suppose the vehicle has some pitch angle, say 25 degrees; then Tan(θ)=Tan(25)=.47 . From equation 4 on page 32

$\dot{\phi}$ = p+SIN(ϕ)*TAN(θ)q+COS(ϕ)*TAN(θ)r

Now, if the vehicle tries some yawing velocity (around z) the value of COS(ϕ)*TAN(θ)*r will be added to $\dot{\phi}$,which means yawing velocity causes some rolling angle of the vehicle when a pitching angle exists in the system. If there were not any pitching angle on the body of the vehicle, having some yawing velocity would not affect the rolling angle. The same logic can be applied to other angles.

Another situation is of interest; suppose the vehicle has some rolling angle ϕ . Then if the vehicle tries some yawing velocity r, pitching angle will change because

$\dot{\theta}$=COS(ϕ)*q - SIN(ϕ)*r

From the equation above it is clear that the existence of rolling angle and yawing velocity will cause some change in pitching angle. The block diagram on page 35 helps one to understand the way that orientation angles are produced.

4.3- TIME-VARIANT PARAMETERS IN THE MODEL

Some parameters of the system are time-variant, and their values depend on the condition of the vehicle. For example, the movement of the manipulator arm on the underwater vehicle changes the moment of inertia of the vehicle. This change depends on the size of the manipulator, its mass and its degrees-of-freedom. If the manipulator does limited maneuvering and its mass is small compared to that of the vehicle, the change in the moment of inertia can be ignored.

4.4- NON-LINEAR TERMS IN THE MODEL

The drag force acting on an underwater vehicle is non-linear. Drag force in any direction is proportional to the second power of the velocity of the body, and also causes hydraulic coupling. In some vehicles the drag force vector is not necessarily parallel to the velocity vector. This usually depends on the geometry of the vehicle. The magnitude of the drag force acting on the vehicle can be modeled as:

$$\text{Drag force} = \rho . Cd . A . v^2/2$$

where:

v=velocity of the vehicle

A=effective area

Cd=drag coefficient

ρ=density of the water

4.5- DESIRED GOALS IN ORIENTATION CONTROL

Orientation control is very difficult to perform manually. Using only open loop manual control the operator is likely to lose control of the vehicle quickly, because she/he cannot decide in sufficient time on the appropriate thruster levels for a particular adjustment in orientation. The orientation controller helps here. It is a servocontroller for thrusters, implemented through an on-line computer. Its function is to help the operator to orient the vehicle.

The orientation controller obtains three angles (ϕ, θ, ψ)from the sensors and produces appropriate voltages to the thrusters. The controller must:

1) Overcome disturbances. It should be noticed that a few types of controllers can overcome disturbances completely. In our design we are interested in a controller that keeps the vehicle within a desired range of orientations, given that some limited disturbance is imposed on the body of the vehicle. The major disturbances are from water flow or manipulator reaction on the body. The rotation of the propellers in different directions might cause some disturbances on vehicle orientation as well.

2) Bring the vehicle to a desired orientation as quickly as possible. (The power of the thrusters and their time constants pose limitations upon the response time .) The reference orientation could be assigned by an operator on the sea surface. The command from the operator can be transmitted to the on - line computer via analog signals generated by a local model. Figure 4.8 shows the local model that helps the operator to assign the reference for the orientation of the vehicle. Of course for relatively complicated tasks the reference orientation for the orientation controller should be assigned by the computer, because the remote human operator cannot re-orient the vehicle with sufficient speed and accuracy. Chapter 4 includes a computer algorithm for a closed loop orientation controller which would replace the operator for complex tasks.

The computer program which solves the equations of motion is named MODEL. MODEL is a general purpose program that can generate almost every dynamic characteristic of the vehicle. MODEL can generate different angular and linear velocities with respect to the global and body axes and determine the location and orientation of the vehicle. Usually most remotely-manned underwater vehicles made for inspection perform tight maneuvers. Consequently they do not work around a special operating point. The operating point of the vehicle changes rapidly and frequently; therefore any set of linearised equations does not simulate the entire modeling exactly. MODEL uses the RUNGE-KUTTA method , order of four, to solve the differential equations. By solving the differential equations by the RUNGE-KUTTA method , the full characteristics of the vehicle about its maneuvering point can be retained. We would like to use MODEL as our plant in our closed loop control system , but to design the controller for MODEL we use another linear model. In the next section we will go through the details of the controller design.

4.6- CONTROLLER DESIGN

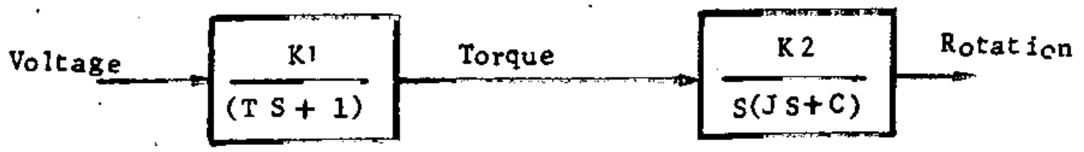
In design of a controller every engineer needs to know the characteristics of the model, for example,pole and zero locations, open loop frequency response, gains, etc. , and especially any nonlinearity parameters.

The non-linear equations of the underwater simulation by themselves are very hard to analyze. There are many classical and modern methods for controller design. Most of these methods need linear models.

The " pole placement + observer" method was chosen for the closed loop orientation control system. This decision was based on the following two considerations:

1)

If the cross-coupling in the model can be ignored, a linearized model for rotation of the vehicle around an arbitrary axis can be written:



T=time constant of the propeller

J =moment of inertia of the vehicle

C =linearized damping coefficient

The model above is not capable of simulating the entire rotation of the vehicle around a particular axis at any time; it is valid around only one operating point. With the help of the model above we would like to analyze the change of the parameters of the model. The open loop system has three poles:

$R1=0$ $R2=-C/J$ $R3=-1/T$

Figure 4.1 shows the root locus of the closed loop system.

For high values of the gain, the system becomes unstable. Two poles of the closed loop system for high values of the open loop gain sit on the right half s-plane and make the system unstable. But for lower values of the open loop gain the closed loop poles sit on a suitable location on the left half s-plane and the system is stable.

The damping force on the vehicle can be expressed as:

$$\rho A C_d V^2/2$$

The following is a linear approximation of the equation above

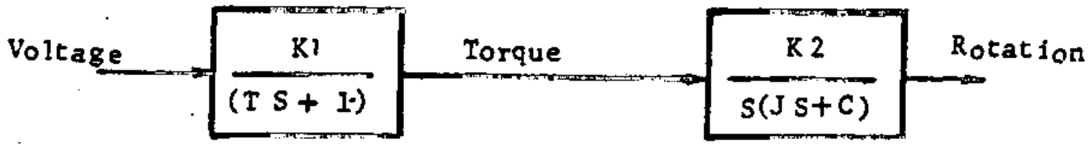
$$C.V$$

C is constant around a given operating point and changes when the vehicle changes its operating point. With inspection one can tell that C increases at higher speed. In general the orientation and velocity of the vehicle have significant effects on the parameters of the vehicle. Figure 4.2 shows the root locus of the closed loop system at different speeds.

At low velocities for most values of the open loop gain the system is unstable. Further we do not even have much control on the open loop gain, because the open loop gain changes and this change depends on the orientation of the vehicle. The change of the parameters in the model causes the closed loop poles to move around. This movement cannot be predicted very precisely. If we use a lead compensator we can pull the roots from the imaginary axis to the left. In other words, this problem by itself can be solved, but other parameters have some contribution to the shape of the root locus too , for example moment of inertia and gain .

2)

We now return to our simple linear model



where: K=K1*K2

The value of the K in the G(S) term

$$G(S) = \frac{K}{(TS+1)(JS+C)\,S}$$

changes because K shows the contribution of a special torque to generate rotation around its axis. If we look at the matrix below we can see that

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \mathrm{SIN}(\phi)*\mathrm{TAN}(\theta) & \mathrm{COS}(\phi)*\mathrm{TAN}(\theta) \\ 0 & \mathrm{COS}(\phi) & -\mathrm{SIN}(\phi) \\ 0 & \mathrm{COS}(\phi)/\mathrm{SIN}(\theta) & \mathrm{COS}(\phi)/\mathrm{COS}(\theta) \end{bmatrix} \times \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

For example, for any pitching angular speed we should get some pitching angle if the rolling angle is not 90°. The value of the pitching angle coming from pitching velocity depends on the rolling angle of the vehicle. The same is true for yawing of the body,i.e. the value of the yawing angle coming from the yawing velocity depends on both rolling angle and pitching angle.

$$\dot{\psi} = \mathrm{COS}(\phi)/\mathrm{COS}(\theta)*r$$

By inspection one can tell that the contribution of any torque around its axis depends on the orientation of the vehicle and can be represented by avarying gain in the simple linear model above. The Bode diagram of the simplified linear model can show the effect of the open loop gain change (see figure 4.3).

If K changes , then the shape of the Bode diagram will not change, but it will be shifted. The shift of the Bode diagram causes the crossover frequency to move. The phase angle diagram does not change either. By changing the open loop gain of the system, the crossover frequency gets bigger and the value of the phase margin gets smaller. Small values of the phase margin cause instability, but this problem can also be overcome by adding some lead angle to the open loop.

Control law + observer is a superior method because it guarantees the location of the closed loop poles even when the items listed below change:

1) drag force
2) moment of inertia of the vehicle
3) the degree of the contribution of a torque around a special angle ,or coupling in the system
4) the values of the command inputs
5) the values of the disturbances coming from the arm or from water flow

The more information we collect for the controllers, the better we can control the vehicle. The only information that we can measure from a real vehicle for orientation control are angles. By using an observer, we can feed the estimated angular velocities to the controller,too. We are interested in design of a controller which guarantees that the vehicle will behave more or less as we want. Estimator design is one of the methods in multi-variable systems that can guarantee the locations of the closed loop poles and the values of the phase margin, even if the model is nonlinear.

H(S) is a linear model which can show the relation between the torque produced by the thrusters and the input voltage to the thrusters. The thrust force produced by a thruster is proportional to the angular velocity of the propeller.

$$\frac{\text{thrust force}}{\text{voltage}} = \boxed{\frac{K1}{TS+1}}$$

T can be calculated from the inertia and drag characteristics of the propeller. Therefore the linearized model for orientation around just one axis can be written in the S domain:

I)
$$\begin{tikzcd} {} \arrow[r] & \boxed{\frac{K1}{TS+1}} \arrow[r] & \boxed{\frac{K2}{S(JS+C)}} \arrow[r] & {} \end{tikzcd}$$

II)
$$\frac{Z-1}{Z} \times \left[\begin{matrix}\text{Z-TRANSFORM}\\ \text{OF}\end{matrix} \left[\frac{K1*K2}{S^2(TS+1)(JS+C)} \right] \right]$$

With the help of the zero-order-hold model, the discrete model in Z domain can be derived.

We should notice that the linearized model is just useful for us to approach our design and is not capable of modelling the vehicle accurately. As will be explained later the linear model derived by equation I is just useful to construct the states of the system for the controller. In other words, we build a linear model that tries to approach the dynamic behavior of the non-linear model (or plant). Because of the non-linearities in the original model we have to correct the linear model almost continuously.

It would be convenient if all of the states that we need for feedback were available. This is not usually possible, because of limited sensors and difficulty in measurement. If all states of the system were available, then we would be able to proceed with the first design step, the "control law ." When all of the states needed for feedback are not available, we are led toward "estimator" design, in order to estimate the complete vector of the system. The final stage of design consists of combining the control law and the estimator, where all control law calculations are based on the estimated states rather than the actual states.

The control law simply means the feedback of all of the states of the system and their multiplication by appropriate gains to form the control signal scalar.

$$u=-K*x=-[K1 \quad K2 \quad K3 \quad \ldots .]\begin{bmatrix} x1 \\ x2 \\ x3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

The digital state equation form of the model can be shown as:

$$X(n+1)=\phi.X(n)+ \Gamma.U(n)+ \Gamma 1.W(n)$$

If we substitute the result in the digital form of the model, we have:

$$X(n+1)=\phi.X(n)- \Gamma K.X(n)+ \Gamma 1.W(n)$$

The characteristic equation of the closed loop system is:

$$\det\left|ZI-\phi+ \Gamma K\right| =0$$

The control law design then can be stated as a calculation of the elements of k so that the roots of the characteristic quation can be placed in desirable locations. A very convenient formula has been derived by Ackermann. This relation is:

$$K=[0\ 0\ \ldots\ldots 0\ 1][\Gamma\ \phi\Gamma\ \ldots\ldots\ldots\ \phi^{n-1}\Gamma]$$

where $[\Gamma\ \phi\Gamma\ \phi^{2}\Gamma\ \ldots\ldots]$ is called the controllability matrix, n is the order of the system and $\alpha_c(\phi)$ is:

$$\alpha_c(\phi)=\phi^{n}-\alpha_1\phi^{n-1}-\alpha_2\phi^{n-2}\ldots\ldots\alpha_n.I$$

where the α_i's are the coefficients of the desired characteristic equation, that is,

$$\alpha_c(Z)=|ZI-\phi+\Gamma K|=Z^{n}-\alpha_1 Z^{n-1}\ldots\ldots\ldots-\alpha_n$$

4.7- ESTIMATOR DESIGN

In control law design we assume that all states of the system are available for feedback. Here we explain the algorithm that constructs all states of the system, given some measured states of the system. The obvious way to construct the states of the system is to make a dynamic model of the vehicle. We present the model in matrix form:

$$X(n+1) = \Phi X(n) + \Gamma U(n)$$

The matrix form of the model is derived from the linear equation II on page 46. Notice that this linear model is only good for estimating the states of the system in the controller. We emphasize again that the linearized model written in matrix form will be used for the estimator controller and will be recorrected every sampling time.

In the equations above x presents an estimate of the actual state, x. The value of Φ and Γ are known, but they change very rapidly. The changes in Φ and Γ have been explained earlier. Because of the change in Φ and Γ we should redesign the controller when there is an appreciable change in Φ and Γ. Once we write an algorithm for design of the estimator for an arbitrary Φ and Γ, then we can use that algorithm to determine the estimator for different values of the Φ and Γ at different orientations of the vehicle.

The beauty of the estimator design is that in a systematic way it enables us to design the controller on-line by computer as a function of changes in orientation and physical configuration of the vehicle. In other words, we teach the computer how to design the controller for thrusters whenever it is necessary to redesign. The dynamics of the vehicle change constantly, and this causes the estimator algorithm to design the servocontroller continually.

$$G_1 = \frac{K1}{TS+1}$$

$$G_2 = \frac{K2}{S(jS+C)}$$

G_1 = linear model for motor made by observer

G_2 = linear model for body rotation made by observer

$$G = G_1 . G_2 = \frac{K1 . K2}{(TS+1)(jS+C)S}$$

$$G = \frac{K1 . K2}{S(Tj S^2) + S^2(TC + j) + CS} = \frac{\theta}{V} = \frac{\text{radian}}{\text{volts}}$$

$$\frac{\theta}{V} = \frac{\frac{K1 . K2}{Tj}}{S^3 + \frac{TC + j}{Tj} . S^2 + \frac{C}{Tj} S}$$

$$\frac{\theta}{V} = \frac{K}{S^3 + BS^2 + AS} \qquad K = \frac{K1 . K2}{Tj}$$

$$\dddot{\theta} + B\ddot{\theta} + A\dot{\theta} = KV \qquad B = \frac{TC + j}{Tj}$$

$X1 = \theta$

$X2 = \dot{\theta}$

$X3 = \ddot{\theta}$

$$A = \frac{C}{Tj}$$

$$\begin{bmatrix} \dot{X1} \\ \dot{X2} \\ \dot{X3} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -A & -B \end{bmatrix} \begin{bmatrix} X1 \\ X2 \\ X3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ K \end{bmatrix} V$$

$$M = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -A & -B \end{bmatrix} \qquad G = \begin{bmatrix} 0 \\ 0 \\ K \end{bmatrix}$$

$$e^{Mt} = \begin{bmatrix} F1 & F4 & F7 \\ F2 & F5 & F8 \\ F3 & F6 & F9 \end{bmatrix} = F$$

$F1 = 1$

$F2 = 0$

$F3 = 0$

$F4 = D1 + B \cdot D3$

$F5 = D2 + B \cdot D1$

$F6 = -C \cdot D1$

$F7 = D3$

$F8 = D1$

$F9 = D2$

$$D1 = \frac{1}{b-a}\left(e^{-at} - e^{-bt}\right)$$

$$D2 = \frac{1}{b-a}\left(b . e^{-bt} - a\, e^{-bt}\right)$$

$$D3 = \frac{1}{ab}\left[1 + \frac{1}{a-b}\left(b\, e^{-at} - a . e^{-bt}\right)\right]$$

$$a = \frac{1}{T}$$

$$b = \frac{c}{J}$$

$$\Phi = I + FT + \frac{F^2 T^2}{2!} + \cdots\cdots$$

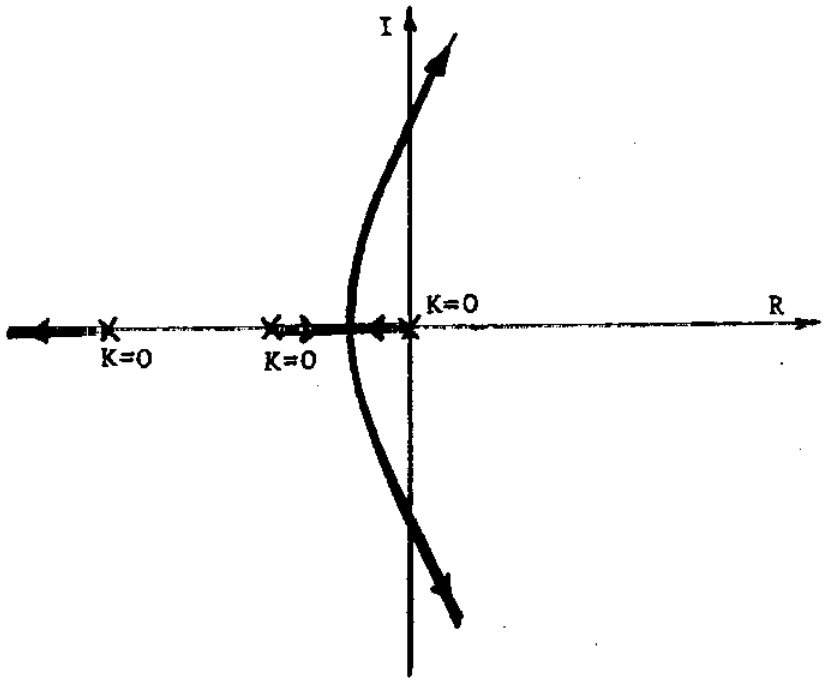
$$\Gamma = \left[IT + F\frac{T^2}{2!} + \frac{F^2 . T^3}{3!} + \cdots\cdots\right] G$$

ϕ and Γ are coefficients of digital model

$$\left.\begin{aligned} x(n+1) &= \phi \cdot x(n) + \Gamma \cdot u(n) \\ y(n) &= H \cdot x(n) \end{aligned}\right] \rightarrow \text{digital form of the instantaneous model made by the observer}$$

$u(n)$ = Voltage to the motors

$y(n)$ = rotation of the body



Root Locus of the closed loop system

Fig.4.1

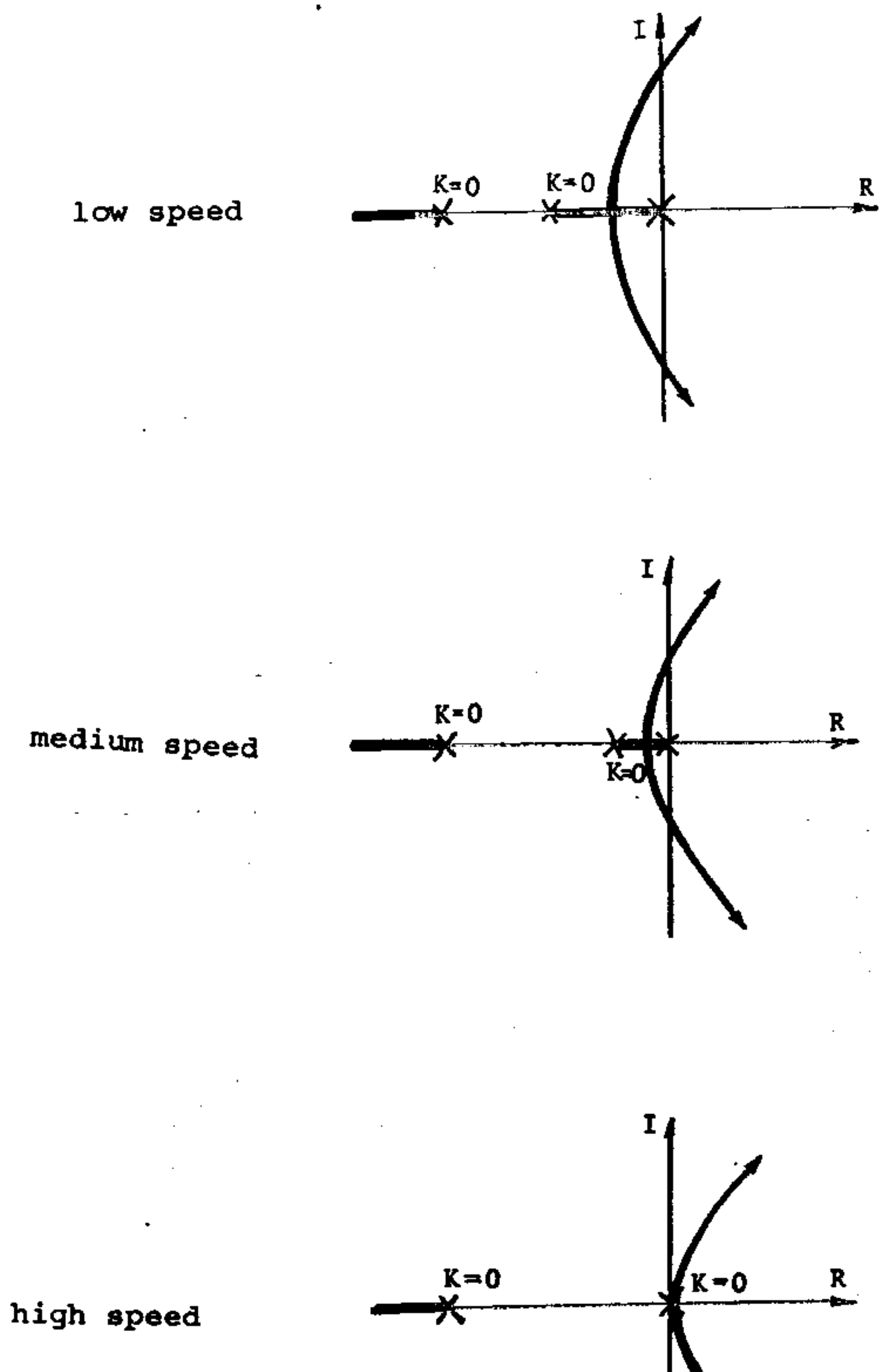
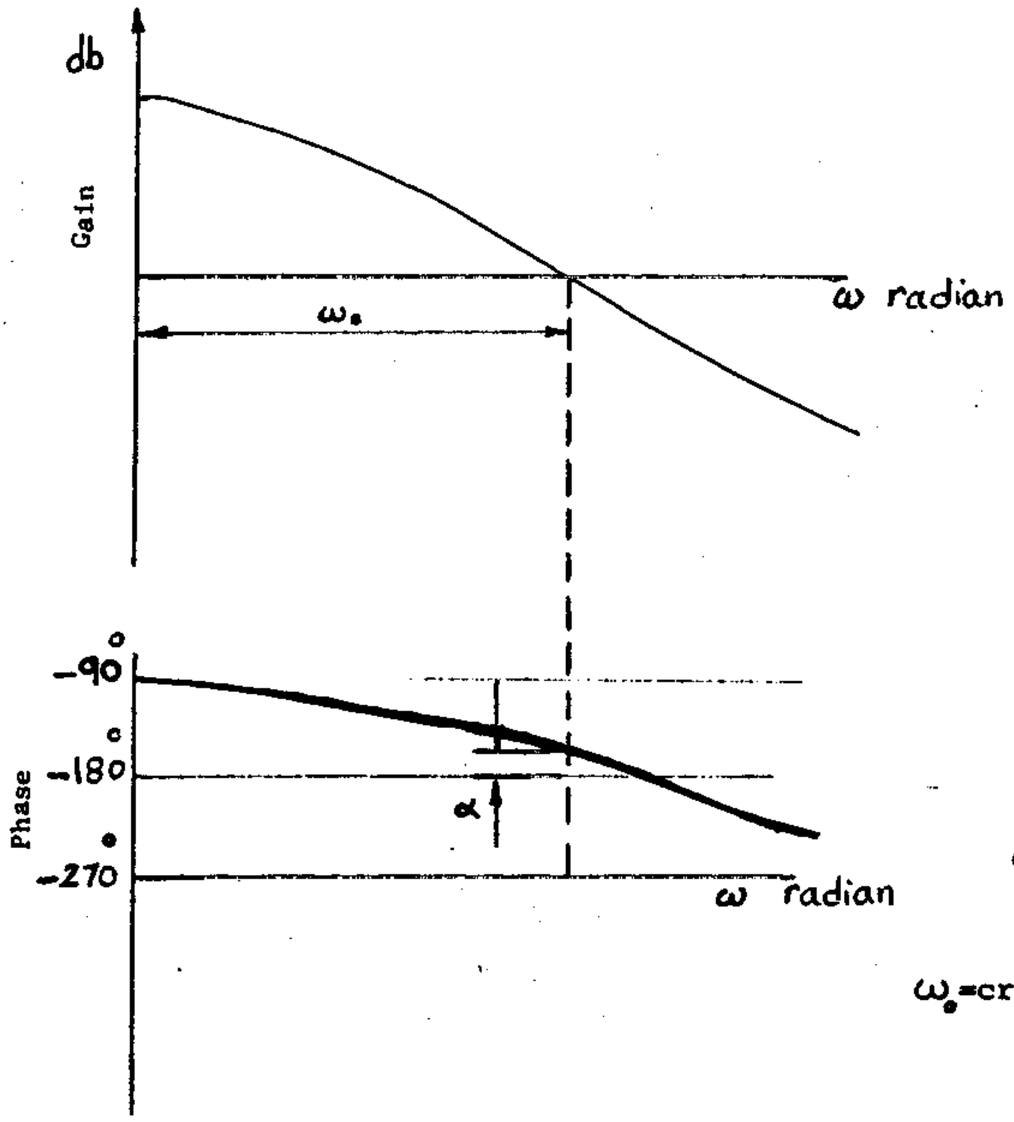


Fig. 4.2 Root locus of the closed loop system at different speed.



α =phase-margin

ω_c=cross-over frequency

Bode diagram of the open loop system

Fig. 4.3

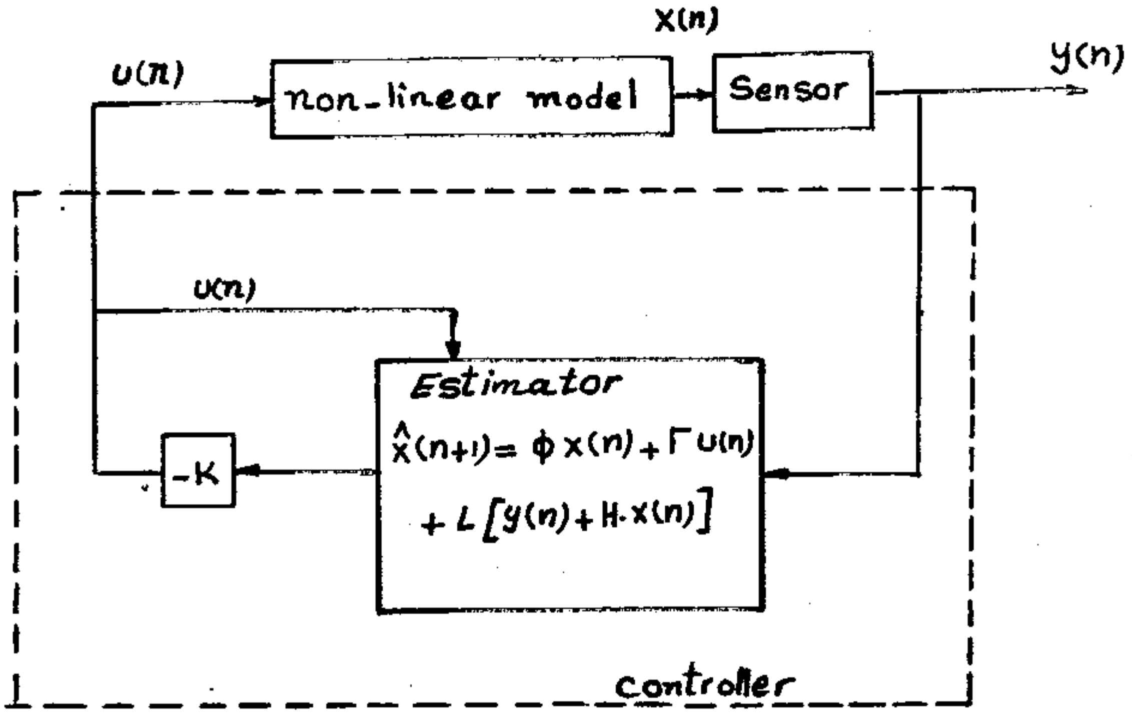

X(n)
u(n)
non-linear model
Sensor
y(n)
u(n)
Estimator
$\hat{X}(n+1) = \phi x(n) + \Gamma u(n) + L[y(n) + H \cdot x(n)]$
-K
controller

Fig. 4.4 : Estimator mechanization for rotation around one axis. Y(n) could be ϕ or Θ or ψ .

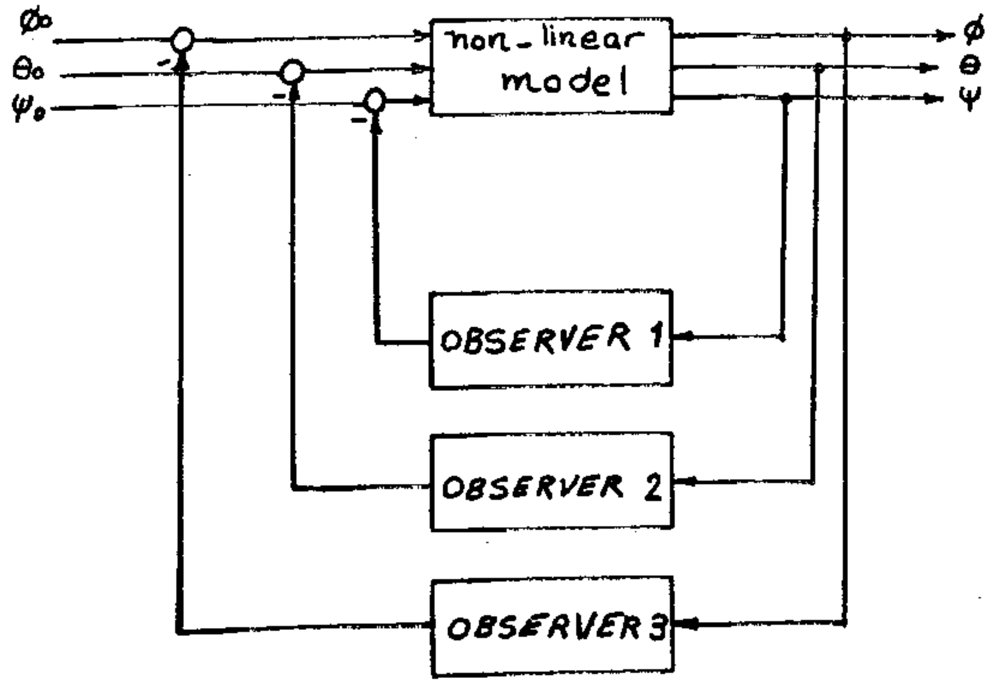


Fig 4.5 : This diagram shows the orientation controller for three different angles.

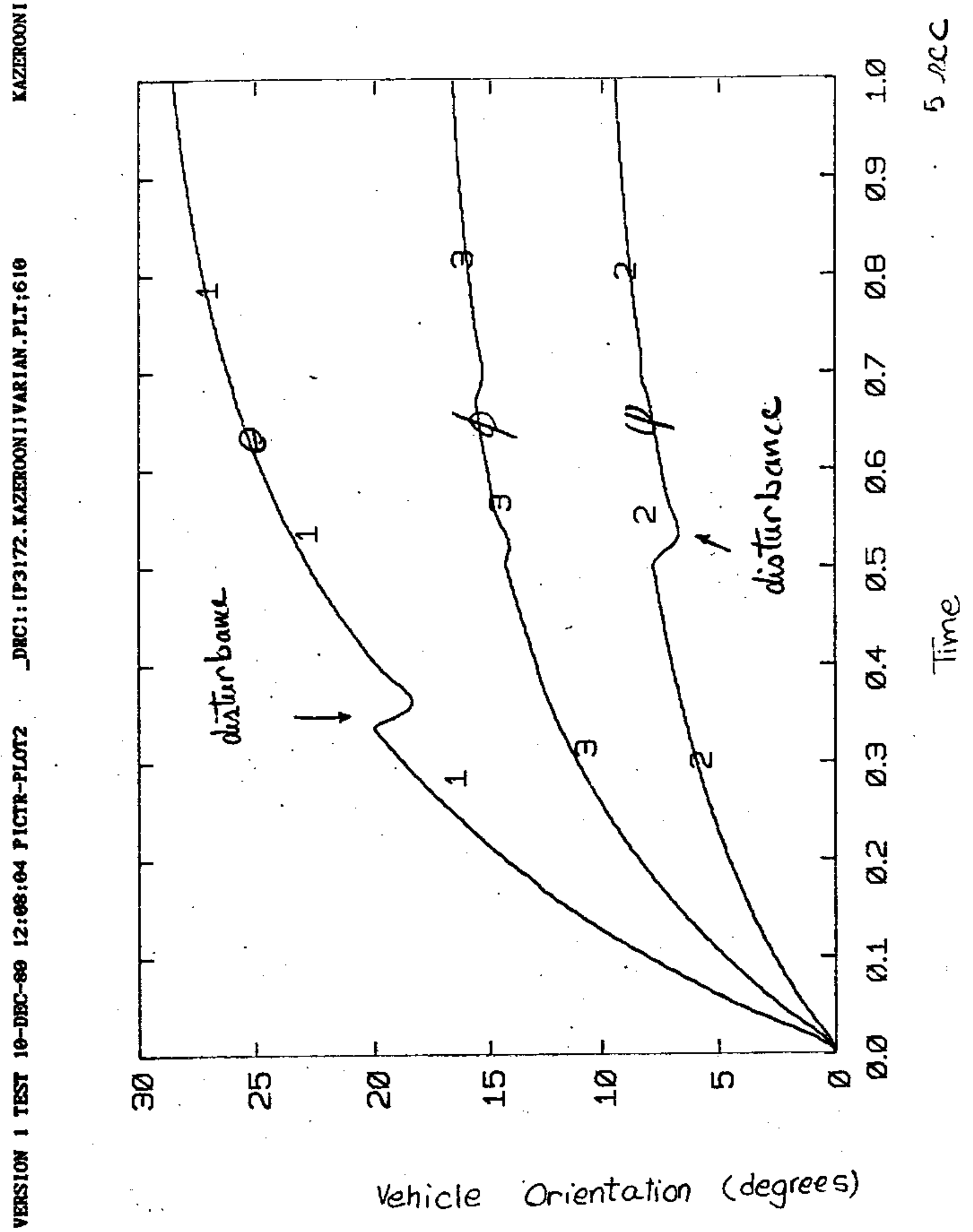


Fig 46 : The controller is able to orient the vehicle in sufficient time. The set point for orientation: Θ_o=30. Φ_o =18. Ψ_o =10.

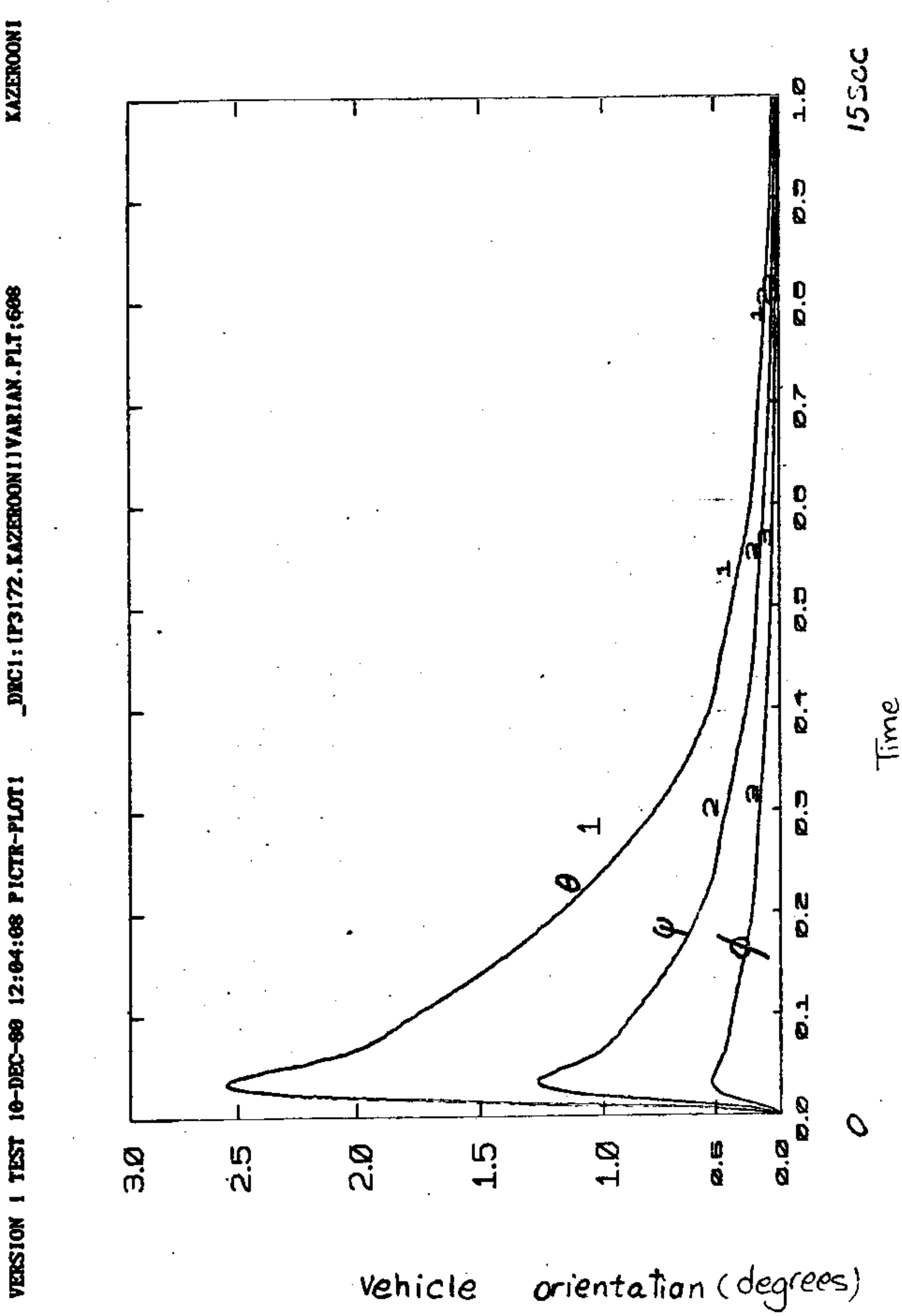


Fig 6.7 : Disturbance rejection property of the controller.

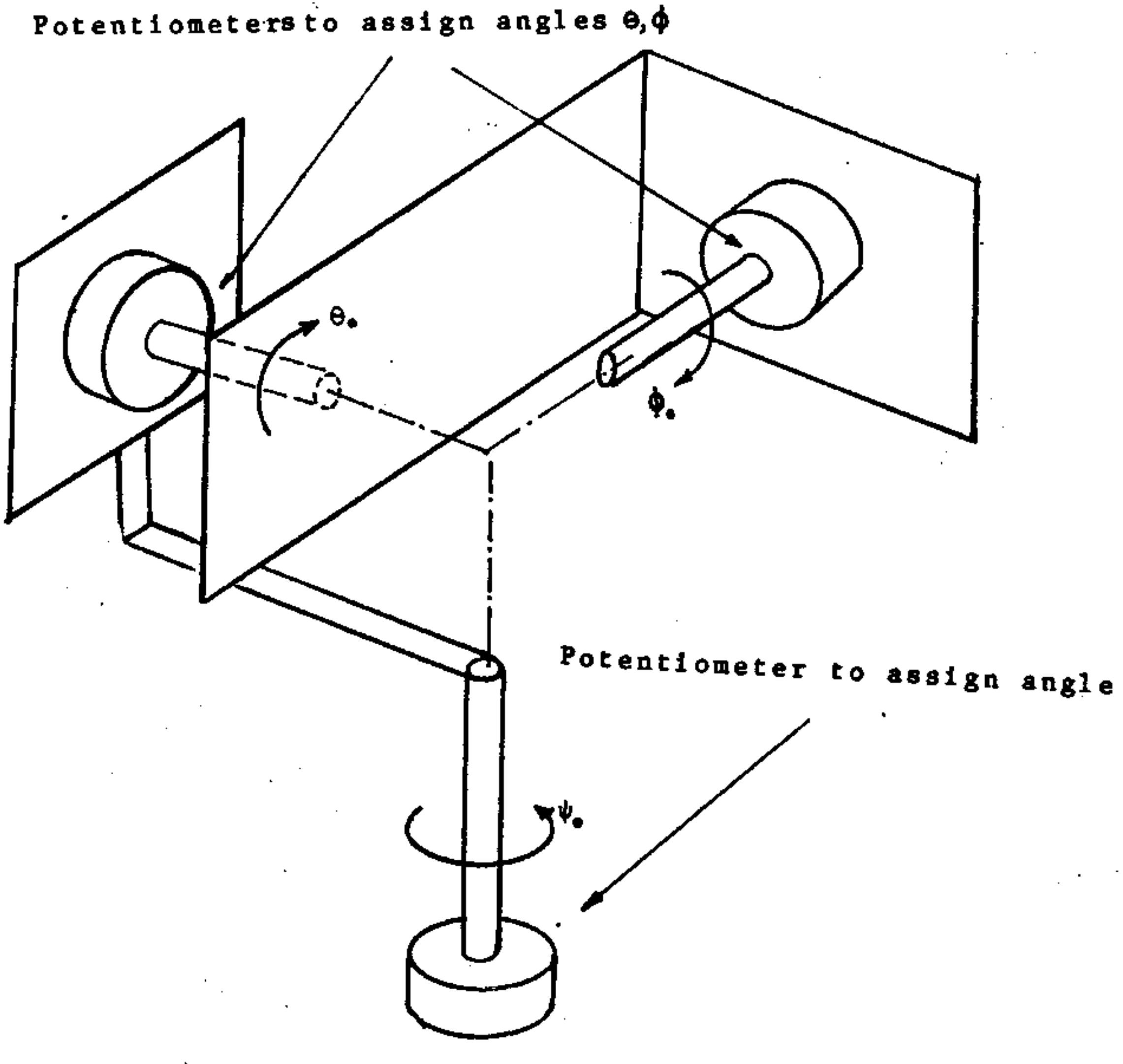

Potentiometers to assign angles θ,ϕ
θ.
ϕ.
Potentiometer to assign angle
ψ.

Fig. 4.8 Local model to assign the orientation angles.

5 POSITION CONTROL

5.1- INTRODUCTION

This chapter explains computer controlled positioning of the remotely manned underwater vehicle in three dimensional space. First the equations for translational motion will be analyzed and then the dynamic characteristics of the translational motion, which affect the controller design, will be studied.

Translational motion of the vehicle does not produce coupling between translational state variables as long as the vehicle does not have any forced rotational motion. Therefore it is strongly recommended that the automatic position control and automatic orientation control not be performed simultaneously.

There are two reasons for this:

1)

Suppose the operator wishes to position the vehicle somewhere under the water with a particular orientation. This is most easily accomplished by locating the vehicle with the position controller while maintaining a fixed orientation, and then orienting the vehicle after it has been positioned. The limitation on the number of the thrusters and their positions with respect to the body of the vehicle makes the simultaneous use of the orientation control and position control almost impossible. In other words, if the vehicle is supposed to move and rotate at the same time to get to a special point with a given orientation, the thrusters' orientation will change because the thrusters are attached to the body with fixed angles. Then it is obvious that the vehicle does not move toward the desired direction anymore because the thrust directions have been changed.

2)

If in some special tasks, it is necessary to rotate and move the vehicle simultaneously, then more thrusters are needed for the vehicle. The vehicle must be redesigned to provide translational movement in a particular direction with different body orientations. The thrusters must be configured so that at least one thruster operates in the desired direction of the translational movement. Most recent designs of underwater vehicles do not include many thrusters. In the few multi-thruster designs the locations of the thrusters on the body are such that the vehicle cannot move and rotate at the same time. Figure 3.1 shows a simple design that allows the vehicle to move in a two dimensional plane with rotation. The controllers designed in this paper do not provide enough control for the vehicles in figure 5.1 ; other servocontrollers with a supervisor program are needed.

With the help of the explanation above we design position controller and orientation controller to be independent of one another in the sense that they do not work simultaneously, and therefore forced rotation and forced translation are truly decoupled.

5.2- NON-LINEAR TERMS IN TRANSLATIONAL MOTION

The only non-linear term in the model is drag force. This force is proportional to the second power of the velocity. Because of the creepwise (no sudden movement) motion of remotely manned underwater vehicles, there is no hydrodynamic coupling between different translational speeds in different directions. The creepwise speed of the vehicle makes the drag force in any direction virtually independent of the speed of the vehicle in another direction.

5.3- COUPLING IN PURE TRANSLATION

Here we explain the details of the coupling that an underwater vehicle experiences in pure translational motion. A vehicle undergoing simultaneous rotation and translation will experience a high degree of coupling, as seen in the equations on page 31 . As explained in the introduction of this chapter we prefer to avoid performing rotation and translation at the same time. The coupling that the vehicle will experience in pure translation depends on the mechanical configuration of the vehicle.

Equation 1 presents the translational motion in the x-direction in the absence of rotation:

$$Tx-Drx-Wgt*SIN(\theta)=M*[\dot{u}+qw-rv-Xg*(q**2+r**2)$$
$$+Yg*(pq-\dot{r})+Zg*(pr+\dot{q})]$$

In most remotely-manned underwater vehicles:

$Xg=0$

$Yg=0$

$Zg\neq 0$

Zg is the vertical distance between the center of mass and the geometric center. If Zg is small, the transient response of the rolling motion of the vehicle is damped and relatively slow. For large Zg, the transient response of the rolling motion of the vehicle is oscillatory. The larger Zg is, the more oscillatory the vehicle is in unforced rolling (the frequency and damping coefficients of oscillation depend on the viscosity of the water as well; however water viscosity is assumed to be constant). The preceding analysis on Zg can easily be checked by inspection of figure 5.2 .

Equation 1 can be simplified:

$$Tx-Drx-Wgt*SIN(\theta)=M*[\dot{u}-rv+Zg*(pr+\dot{q})]$$

Because of the pure translational motion :

p=0

r=0

Rearranging:

$$Tx-Drx-Wgt*SIN(\theta)=M*[\dot{u}+Zg*\dot{q}]$$

The forces in the x-direction produce two components of acceleration:

u=translational acceleration in the x direction

q=rotational acceleration around the y axis

u moves the vehicle toward the x direction. Rotational acceleration around the y-axis , q ,is transient because it is quickly overcome by the moment produced by the force of buoyancy. Once the vehicle reaches its steady state speed in the x direction both q and u die and the vehicle moves smoothly in the x direction.

Figure 5.2 indicates that an unwanted pitching is produced while the vehicle accelerates in the x direction. This is a soft coupling that does not affect the entire positioning task. If Zg=0 this coupling disappears and no pitching angle is observed. If Zg is large, the pitching motion produced will be oscillatory but will eventually die out.

There is similar coupling in the y-direction, (i.e. if the vehicle tries to move in the y-direction and $Zg \neq 0$, some rolling angle ϕ will be produced). The length of Zg plays an important role in the rolling stability of the vehicle. If Zg is large, there is less coupling, because Zg is the moment arm that increases the force of the buoyancy If the moment arm is short (i.e. the center of mass is almost coincident on the geometric center), then translational movement will not produce rotation, but the vehicle will be very sensitive to any rotational force and will turn over with a slight disturbance.

For every underwater vehicle there is an optimum length for Zg. The RCV-150 undersea robot has Zg= 3' (see page 122) which makes an effective passive controller for rolling motion of the vehicle. The non-zero value of Zg in the RCV-150 produces a slight coupling in the translational motion of the vehicle but does not affect the position control extensively.

5.4- DATA TRANSFORMATION

Typically a cable links the robot to a mother ship on the surface. Two microcomputers (one aboard the robot, the other on the mother ship) handle virtually all communications, display, and control functions. The microcomputer in the control station scans the operator console at some given frequency (20 Hz for example). When the operator enters a maneuvering command, the on-surface micro encodes the command and transmits it as an RF-telemetry signal to the robot. The robot's micro then interprets the command and activates the thrusters. The basic controllers such as the position controller and the orientation controller and any supervisor algorithm such as the pathfollower can be written on the surface micro while the robot's micro simply interprets the commands and then activates the thrusters.

The position of the vehicle with respect to the mother ship can be detected roughly by sonar sensors on the mother ship. Having processed the data obtained from the sonar sensors the appropriate controlled variables can be sent to the robot's micro. Data communication systems for various undersea robots are different and depend on distance, angle to the surface and task factors.

5.5- DESIRED GOALS IN POSITION CONTROL

Without automatic position control it is difficult for the operator to position the vehicle. Assigning a sequence of different locations in a cartesian coordinate system is the most difficult task that the operator can perform in positioning the vehicle in open loop operations. In any open loop positioning task the water flow disturbs the vehicle and the operator may lose control of the vehicle. Open loop control of undersea robots can be useful only for simplified tasks in calm water.

The position controller is designed to meet two desired characteristcs:

1)

Positioning the vehicle in three dimensional space as quickly as possible. On page 69 different control criteria for position controller design were discussed. The power of the thrusters is a constraint in achieving this goal. The position reference can be assigned by the operator on the mother ship with the help of analog signals on a console.

2)

Overcoming the disturbances (such as water flow) that cause the vehicle to move from its assigned position. This is an important feature that an open loop controller cannot provide. Again, the power of the thrusters is important. More powerful thrusters with lower time-constants can overcome disturbances more quickly than others. At times, thrusters cannot completely overcome disturbances.

5.6- CONTROLLER DESIGN

The design of the position controllers uses the observer method. The only time-varying non-linear term in translational motion is drag force. The drag force changes with the speed of the vehicle. Because of the existence of time-varying parameters in the model, the position controller should be designed to operate differently at different operating points. An observer is designed based on the simplified linear model that simulates the dynamic behavior of the vehicle. Whenever the dynamics of the vehicle change, the algorithm designs a different observer based on the latest simplified linear model of the vehicle. The position controller has the same structure that the orientation controller has.

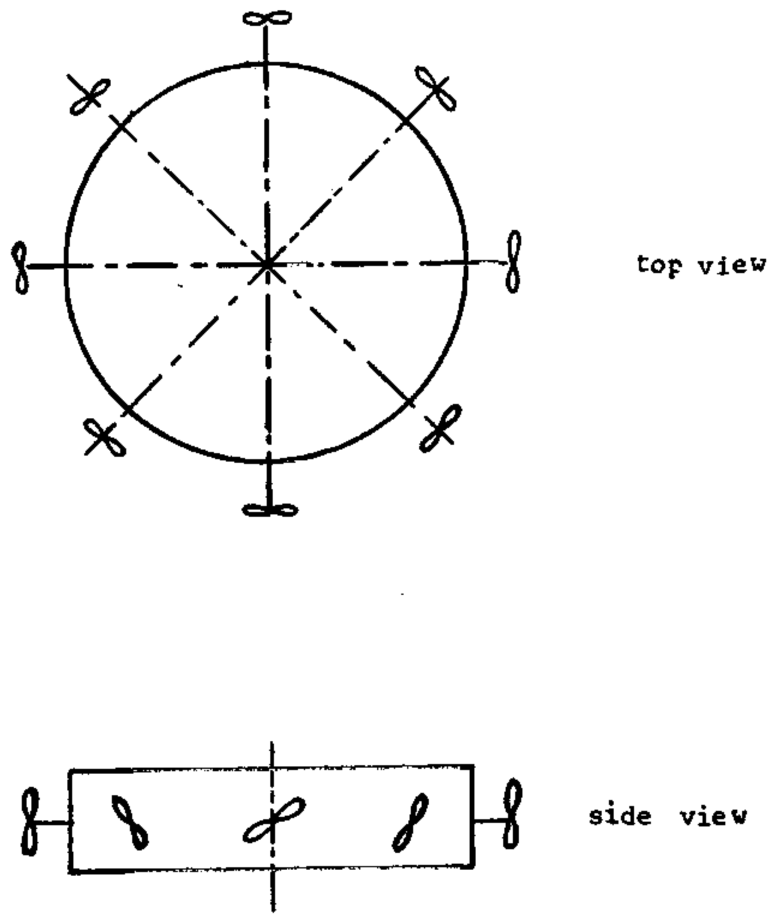


Fig 5.1 A vehicle able to move and rotate at the same time.

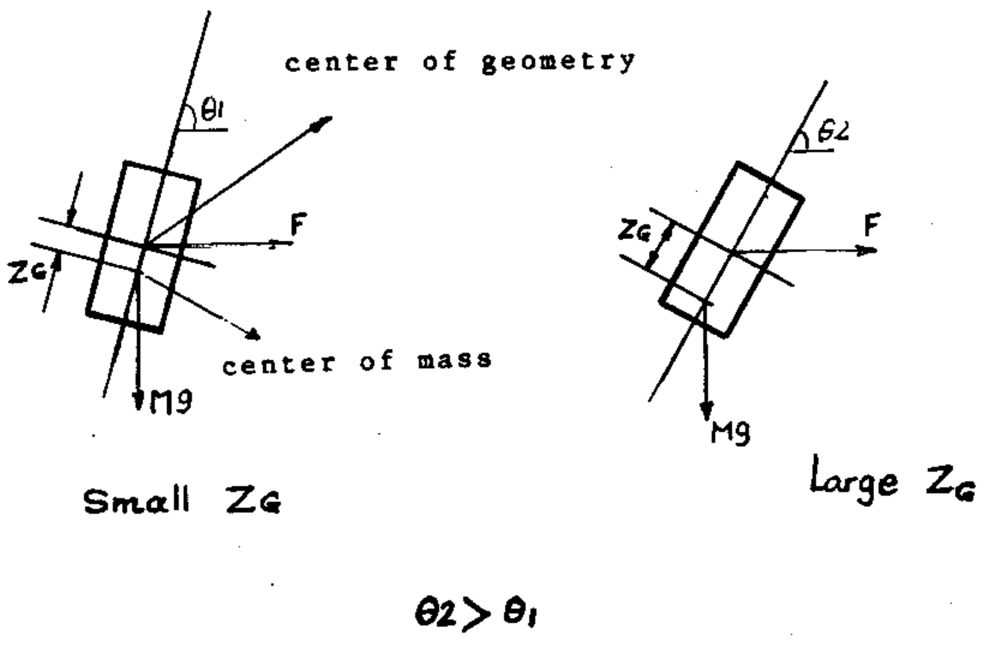


Fig. 5.2 Forward motion causes pitching angle when $Z_g \neq 0$

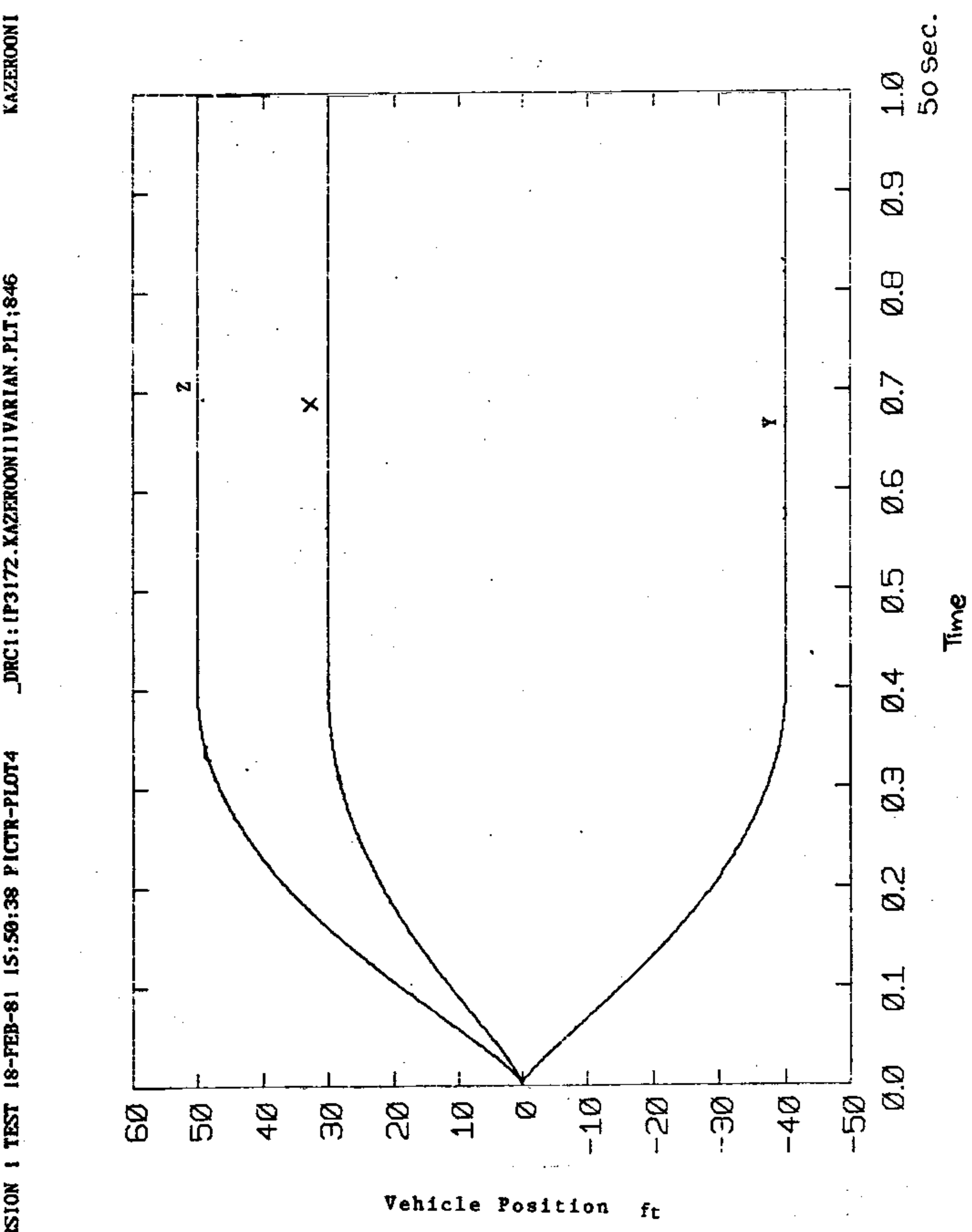


Fig. 5.3 Response of the vehicle to set point assigned for the position controller. $R=30\hat{I}-40\hat{J}+50\hat{K}$

6 BOTTOM-FOLLOWING WITH CONSTANT PITCH ANGLE

6.1- INTRODUCTION

A bottom-following algorithm enables a vehicle to follow an arbitrarily specified path under the water at a small, constant distance from the ocean bottom. The operator brings the vehicle to a certain depth, then turns the vehicle in the horizontal plane to a desired orientation.(That is, the operator assigns some yaw angle to the vehicle.) The operator uses the orientation controller to determine the yaw angle. This is accomplished by a local orientation model.

The operator initiates the bottom-following program, and the vehicle begins moving along the specified yaw angle, following the contour of the bottom at the assigned distance. As long as the vehicle is in automatic bottom-following mode, the orientation controller should remain in effect in order to maintain the vehicle at its specified yaw angle. If the operator changes the yaw angle of the vehicle, the bottom-following algorithm will continue along the new yaw angle.

The pitch angle of the vehicle does not change during the maneuvering; the x-axis of the vehicle is always parallel with the water surface regardless of the contour of the ocean bottom. This algorithm is suitable for vehicles which are not capable of pitching. In some vehicles, because of a large distance between the geometric center and the center of mass, pitching requires tremendous power.

6.2- BOTTOM-FOLLOWING ALGORITHM

The bottom-following algorithm is a supervisor program that operates different servocontrollers at the right times. Of course, a very experienced operator might be able to follow a special path under the water by assigning different reference inputs to the controllers, but she/he could not easily make decisions in sufficient time for complicated tasks like bottom-following.

This algorithm obtains all its necessary information about the environment from sonar sensors. It interprets this data and decides when and how to utilize the thrusters.

The algorithm enables the vehicle to go in any direction while maintaining a constant orientation, and without colliding with the ocean floor. The algorithm uses the position controller to maintain the distance between the vehicle and the bottom at the assigned value; it uses the velocity controller to achieve the desired speed along the path. The desired distance from the bottom and the desired velocity along the yaw direction should be assigned by the operator and can be changed by him even when the vehicle is in the bottom-following mode.

As long as the vehicle is in the automatic bottom-following mode, the orientation of the vehicle does not change unless the operator so decides.

The vehicle is equipped with four sonar sensors,A,B,C and D seen in figure 6.1 . A and B measure the forward and rearward distance, while C and D measure the distance between the vehicle and the ocean bottom. The more information about the environment collected for the algorithm, the better the vehicle can follow its assigned path. To insure that the vehicle recognizes the environment, all paths can be divided into three categories:

1) Paths with slopes between 45° and 90°
2) Paths with slopes between -45° and 45°
3) Paths with slopes between -90° and -45°

Dividing the contour of the ocean bottom into three categories is essential. Categorization of the paths incorporates knowledge into the program that facilitates its recognition of the environment and simplifies the decisions that must be made to drive the servocontrollers. The decision-making of the bottom-following algorithm is based on the information that it obtains from the sonar sensors. Some pre-classified knowledge about the environment stored in the algorithm makes the bottom-following program able to accommodate the type of the environment and therefore to make logical decisions about driving the servocontrollers.

6.3- DECISION-MAKING LOGIC

If in figure 6.1 the measured distance from sonar A is smaller than those from the other sonar sensors, then the environment is considered of type 1 and the vehicle is closest to the bottom at point A. In any path of type 1 there is always some danger that the vehicle will collide at its bow. If the measured distance from sonar B is smaller than those from the other sonar sensors, then the environment is considered of type 3 and point B is the closest part of the vehicle to the ocean bottom. Figure 6.2 shows the three different categories of the environment contours.

The following shows the logic used to realize the environment.

$$\left.\begin{array}{l} AA'<BB' \\ AA'<CC' \\ AA'<DD' \end{array}\right\} \longrightarrow \text{Type } 1$$

$$\left.\begin{array}{l} BB'<AA' \\ BB'<CC' \\ BB'<DD' \end{array}\right\} \longrightarrow \text{Type } 3$$

$$\left.\begin{array}{l} CC'<AA' \\ CC'<BB' \\ DD'<AA' \\ DD'<BB' \end{array}\right\} \longrightarrow \text{Type } 2$$

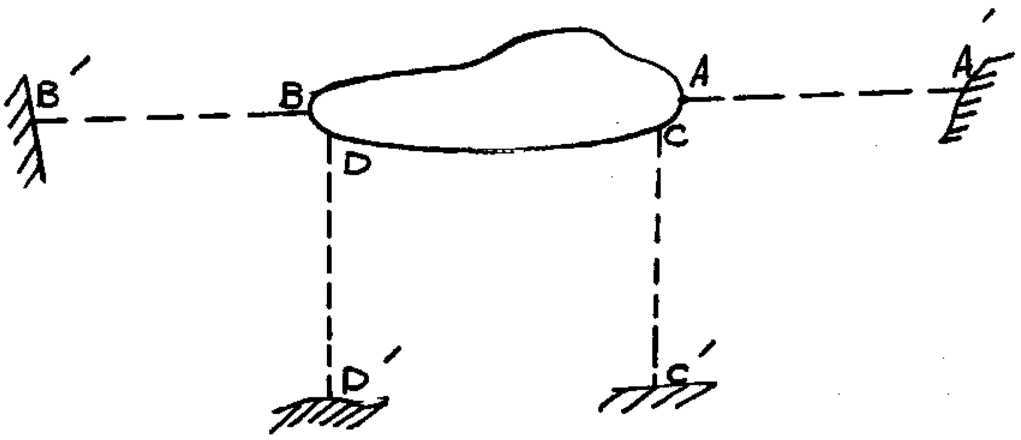
If the vehicle is in an environment of type 1, then the algorithm realizes that the measured distance from sonar A is the smallest measurement. The vehicle then uses its horizontal thrusters to keep AA′ in the assigned range. The horizontal thrusters are run by the position controller. Figure 6.4 shows the maneuvering of the vehicle in an environment of type 1.

If the vehicle moves closer to or farther from the bottom because of a disturbance (for example ocean currents), then the position controller will overcome the disturbance and will bring the vehicle to the desired distance. One must use vertical thrusters to vary the speed along a type 1 path such as that shown in figure 6.3. The control algorithm uses the speed controller to operate the vertical thrusters.

Vertical movement of the vehicle by means of the vertical thrusters and speed controller causes the forward distance to change, which in turn can act like a disturbance for the position controller. This disturbance must be overcome by the position controller. In other words, there can be some mutual forcing or cross-coupling by the control system.

If the vehicle is in an environment of type 2, the vertical thrusters are run by position controllers to keep CC′ or DD′in the desired range. The horizontal thrusters are operated by the speed controller. Any horizontal movement causes a change in the vertical distance CC′ and DD′ , which can be overcome by the position controller.

In an environment of type 3 , the algorithm works the same as it does in type 1.



$$\left.\begin{array}{l} AA'<BB' \\ AA'<CC' \\ AA'<DD' \end{array}\right\} \longrightarrow \text{Type } 1$$

$$\left.\begin{array}{l} BB'<AA' \\ BB'<CC' \\ BB'<DD' \end{array}\right\} \longrightarrow \text{Type } 3$$

$$\left.\begin{array}{l} CC'<AA' \\ CC'<BB' \\ DD'<AA' \\ DD'<BB' \end{array}\right\} \longrightarrow \text{Type } 2$$

Fig. 6.1 Position of the sonars on the vehicle

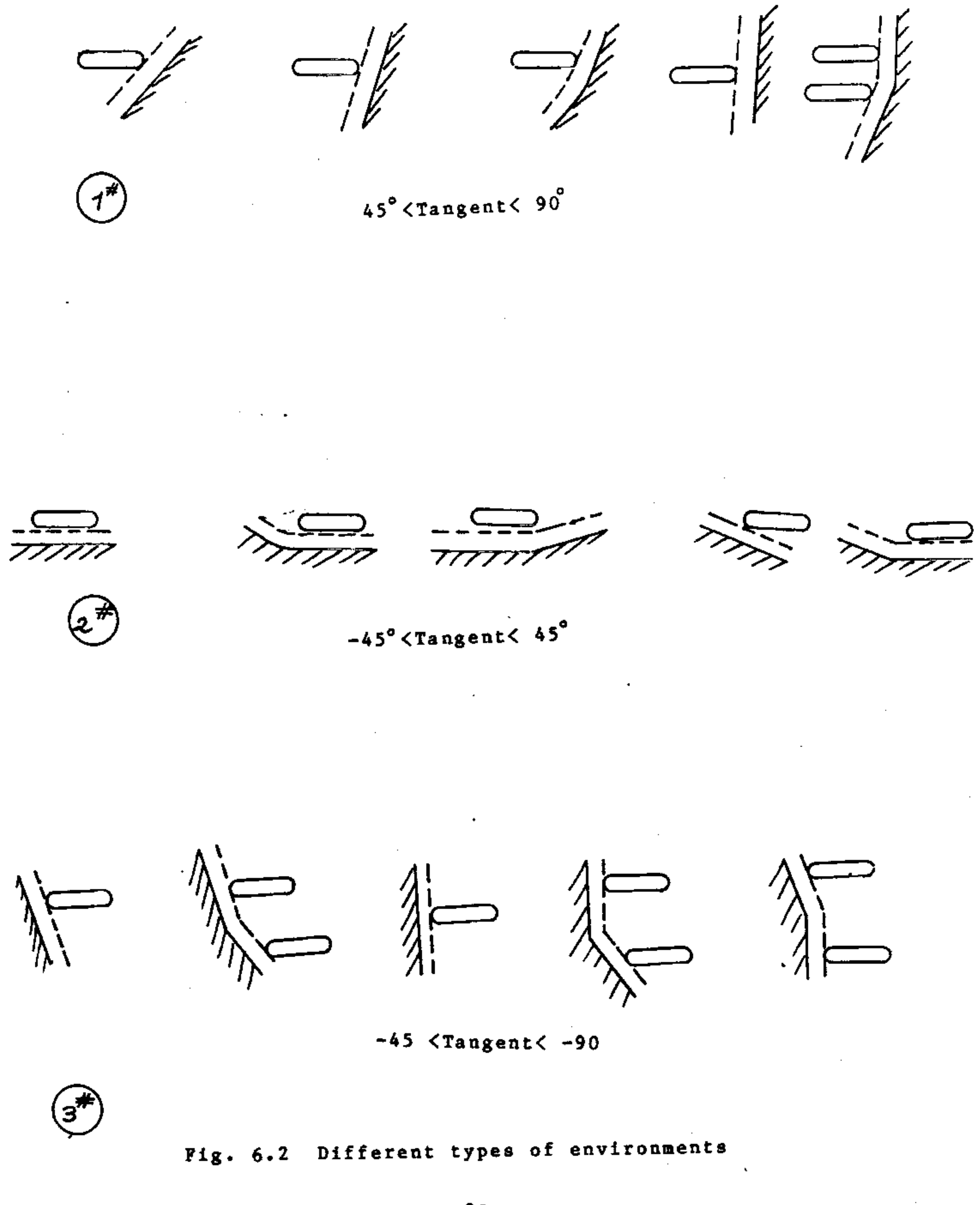


Fig. 6.2 Different types of environments

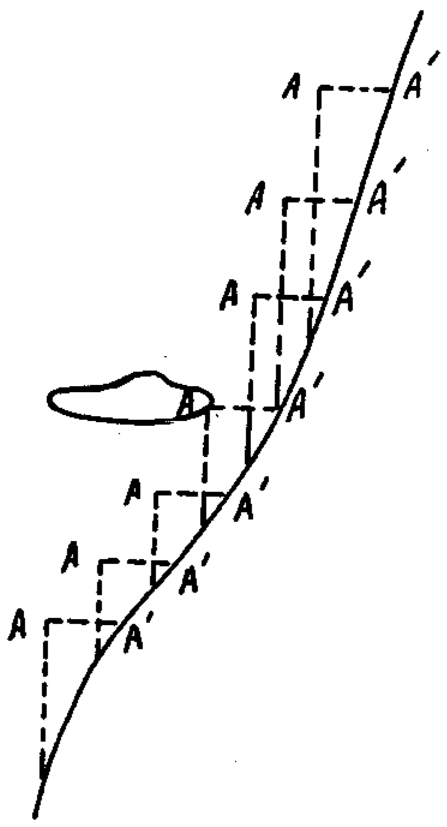


Fig. 6.3 Locus of the vehicle in following the path of type #1

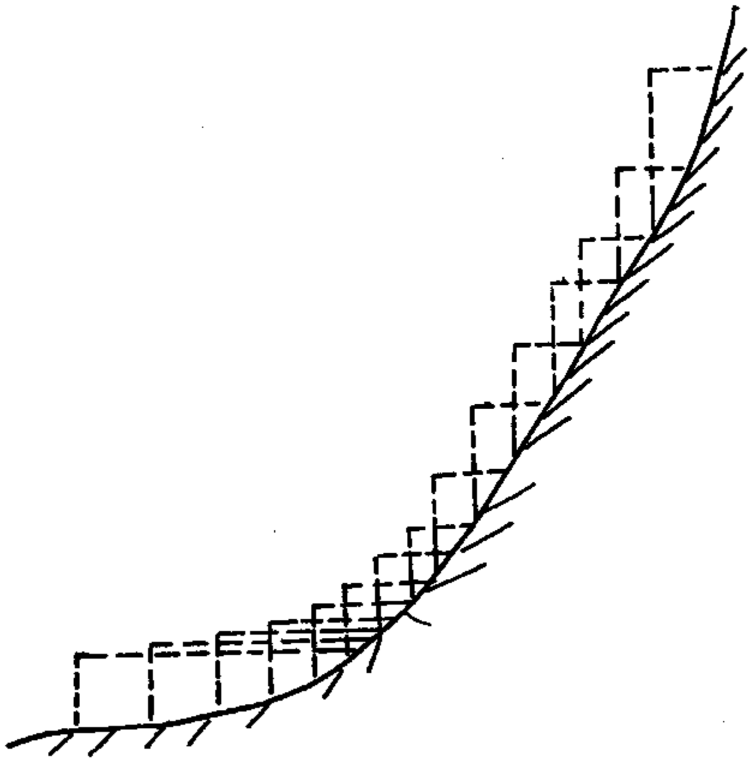


Fig. 6.4 Locus of the vehicle in following the a transition path from type #2 to type #1

7 COMPUTER CONTROL AUTOPILOT FOR ALVIN

7.1- INTRODUCTION

In this chapter the design of a special purpose autopilot is described for Alvin. Alvin is a heavy submarine which belongs to Woods Hole Oceanographic Institution; it is equipped with scientific research instruments. One of the important tasks that Alvin frequently performs is following the bottom of the ocean at a constant distance with constant orientation. This task is currently being performed by a human pilot. Alvin is equipped with one sonar that scans horizontally. The sonar currently on the submarine helps the operator to realize the contour of the ocean bottom.

The objective of a computer controlled bottom following autopilot is to provide, through coupled computer software, the ability to command the thrusters based on the information which is collected by the sonars. However, Alvin is not able to follow any arbitrary path under the water because of the following constraints:

1) Alvin has three thrusters which are not powerful relative to Alvin's weight. More thrusters at different directions would increase the ability of Alvin to maneuver

more tightly under the water.

2) Alvin's huge inertia puts some limitation on its maneuvering relative to a turbulent or strong cross-current environment. In other words, small submarines with powerful and fast thrusters will perform better automatic tasks.

Because of hardware limitations on Alvin (weak thrusters, huge inertia, limited data aquisition systems, big size), the majority of autopilot tasks should be performed by software. In other words some data analysis, prediction and decision-making should be done by an on-line computer, prior to any hardware functioning.

7.1- AUTOPILOT SIMULATION ON PDP11/34

The real time autopilot simulation for Alvin in the Man-Machine System Laboratory consists of three routines that communicate with each other with the help of the buffer in the computer. These routines are called STERN, TANG and ALVIN. Every routine contains some subroutines to perform each part of the simulation. These routines are synchronized and run in real time. Figure 7.1 shows the block diagram of the autopilot simulation.

STERN

This program simulates the dynamics and kinematics of all the hardware that exist on the submarine. It represents the dynamics of the submarine, kinematics and dynamics of the sonars, and graphical simulation of the submarine, environment and sonar beams. A vector display terminal is employed to present the graphical simulation. This program consists of different subroutines that simulate different parts of STERN. The following describes the different subroutines.

1) DYNAMIC SIMULATION

This routine simulates the dynamic behavior of the submarine. The dynamic equations of the underwater vehicles have been analyzed in chapter 3 carefully. In design of the autopilot it is not necessary to analyze the entire dynamics of the submarine, however the equations on page 31 can be applied to Alvin too. Since we are interested in the bottom following aspect of the submarine in two dimensions with constant orientation some simplifications on the model can be done.

The distance between center of mass and center of gravity is 22 inches. The heavy weight and long distance between center of mass and center of gravity causes Alvin to have hardly any pitching or rolling. Figure 7.2 simply shows that the passive control of pitching and rolling will cause the submarine to reject any rolling or pitching disturbance on the body of the submarine. If the submarine should not have any passive control on rolling and pitching, an active orientation control should be used to keep the submarine in zero orientation, because the entire function of the autopilot for Alvin is designed based on zero orientation. The objective of the autopilot is to follow the bottom of the ocean in two dimensions, in other words the autopilot will not be able to assign any yaw angle to the submarine. The yaw angle to the submarine should be assigned by the operator with the help of the orientation controller.

With regard to the explanation above, we can ignore the entire rotational motion of the submarine in all degrees of freedom. This simplification leaves the model just with translational equations in two dimensions. The thrusters on the submarine can be modeled like a first order system with small time constant.

2) GRAPHICS

This routine presents a graphical simulation of the underwater vehicle, sonar beams and four-thousand foot environment on a vector display terminal. The screen has been divided into two parts. The upper part shows a close-up picture of the submarine and two hundred feet of environment. In other words the upper part of the screen looks like a moving picture taken by a camera which moves with the submarine. The distance between the camera and the submarine is such that two hundred feet of environment can be seen on the vector display terminal. The lower part of the screen shows the submarine and four thousand feet of environment. This part of the screen looks like a picture taken by a fixed camera which is located on the floor of the ocean far from the submarine. Therefore four thousand feet of the environment can be viewed on the vector display terminal. With two pictures of the submarine on the screen, one can see the behavior of the submarine with respect to the environment under different control modes. Figure 73 shows the arrangement of the graphical simulation on the vector display terminal. A copy of the graphical simulation program is appended.

3) SONARS

This subroutine simulates the function of the sonar system. By applying different parameters to this program such as sweeping angle, period of motion, number of beams per second, and effective length of the beam one can simulate different sonars on the submarine. This routine gives the operator the capability to test different sonars for the submarine. For the autopilot design task, two sonar systems which scan vertically are needed. The sonar simulation allows for different rotational motions for the

the sonar, too. The rotational motion of the sonar is compeletely under control and different motions can be assigned. Figure 7.4 shows different motions of the sonar. Given the angle at which the sonar beam points toward the environment, the control system calculates the intersection of the sonar beam with the environment and records the length of the beam and the angle of the beam relative to the submarine. The angle of the sonar relative to the submarine is the real time input to the control routine and the length of the beam is the output. Figure 7.5 shows schematically the input and output of the routine. The routine is synchronized by a real time clock. Both sonars on the submarine have the same dynamics and specifications but they are commanded differently. The first sonar scans mechanically without having any command from any software, but the second sonar angle relative to the submarine is assigned by TANG.

4) MOTOR

This routine simulates the kinematic behavior of the motor which runs the first sonar. The dynamics of the motor are not important because the motion of the first sonar is a steady state periodic function, and once the type of motion of the sonar is decided, for the entire maneuvering the same motion repeats itself. Figure 7.4 shows the different motions for the first sonar. The angle of the first sonar at any instant of time is assigned by this subroutine. A copy of this subroutine is appended.

TANG

This routine is the heart of the autopilot software. It fetches the data from the first sonar and after some analysis with its subroutines it makes decisions about different tasks. The communication between this routine and other routines is done by the buffer in the computer. The angle of the first sonar [Θ] relative to the submarine and the length of the beam [L] are the real time inputs to this routine. This routine receives new values for L and θ when the first sonar receives new data. After every period of the motion of the first sonar (when θ equals -90 degrees), the approximate value for the tangent of the environment is calculated.

The tangent for the environment is calculated from the equation:

$$\mathrm{TAN}(\alpha) = \frac{\sum XY - \frac{\sum X \sum Y}{n}}{\sum X \quad \frac{(\sum X)^2}{n}}$$

$$X = L*\mathrm{COS}(\theta)$$

$$Y = L*\mathrm{SIN}(\theta)$$

Subroutine MATH calculates the angle for the second sonar based on its tangent to the environment. The measured distance by the second sonar is the set point for the servocontroller. Any measurement from the second sonar greater than forty feet cannot be trusted. Therefore MATH tries to assign an angle for the second sonar such that the measured length from the sonar to the environment is around forty feet. Of course the function of this subroutine depends on the specification of the sonar system and type of environment. Figure 7.6 shows the the relation between the assigned angle for the second sonar and other parameters. The assigned angle for the second sonar is one of the outputs of the this routine.

ALVIN

This program is the servocontroller for the vertical thrusters. The set point for the servocontroller comes from the second sonar. The output of this software is the voltage to the vertical thruster. Because of physical equipment, the hydraulic motor must turn at constant speed; no amplitude modulation technique is now compatible with the autopilot software. Thus, in order not to have to change the whole thruster system, the bangbang controller with dead zone and hysteresis was chosen for operating the thrusters. Figure 7.7 shows the block diagram of the servocontroller. The value of the dead zone and hysteresis are assigned from TANG. This is another decision that TANG makes based on the reconstruction of the environment. Figures 7.8 to 7.19 show the performance of the ALVIN autopilot simulation in several environments.

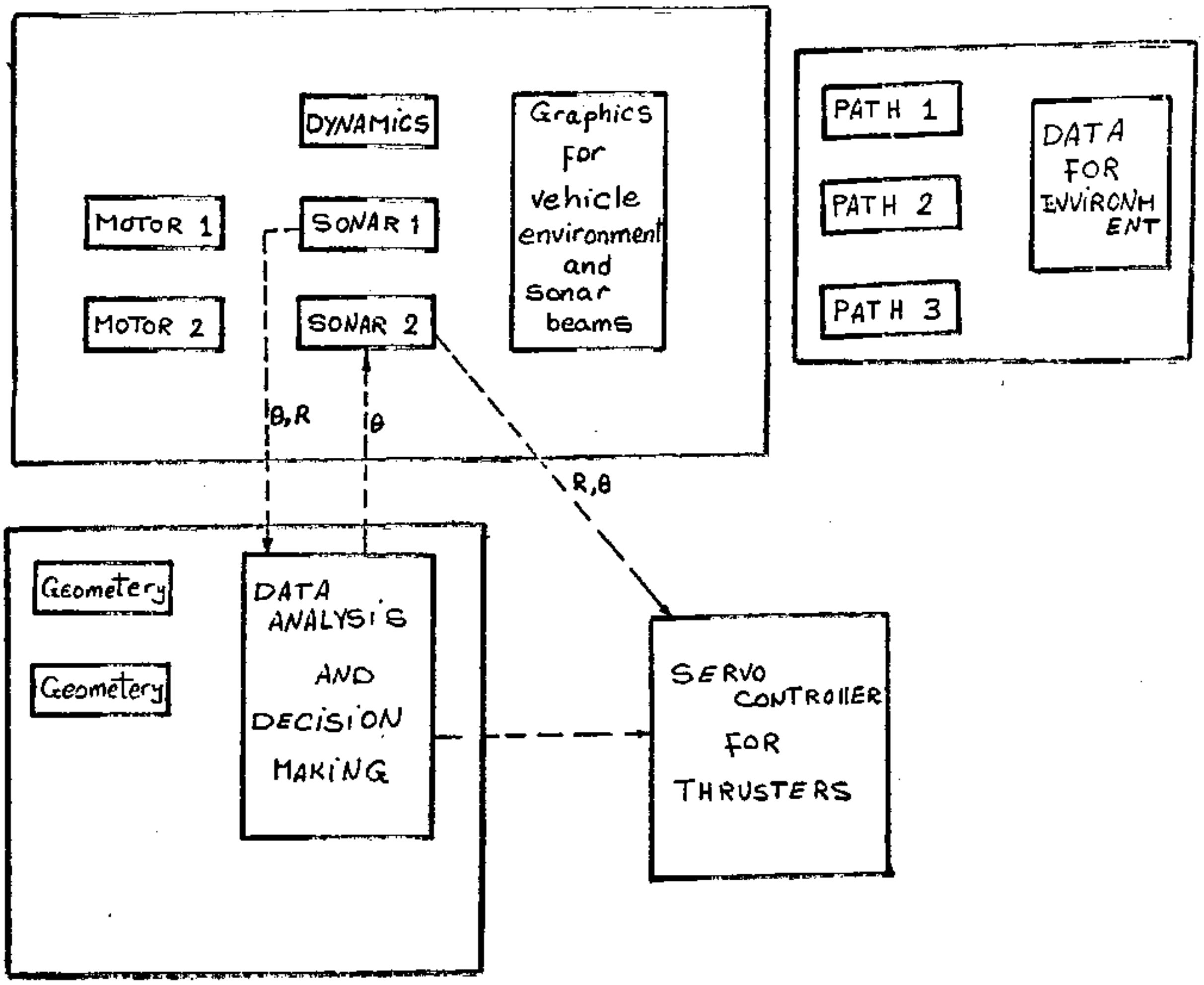

DYNAMICS
Graphics for vehicle environment and Sonar beams
MOTOR 1
SONAR 1
MOTOR 2
SONAR 2
θ,R
θ
R,θ
PATH 1
PATH 2
PATH 3
DATA FOR ENVIRONMENT
Geometery
Geometery
DATA ANALYSIS AND DECISION MAKING
SERVO CONTROLLER FOR THRUSTERS

Fig. 7.1 Diagram showing the autopilot simulation.

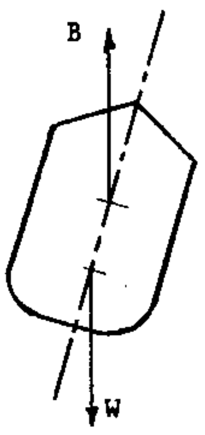


Fig. 7.2 The distance between the center of mass and the center of geometry produces a strong passive control on the pitching and rolling angle.

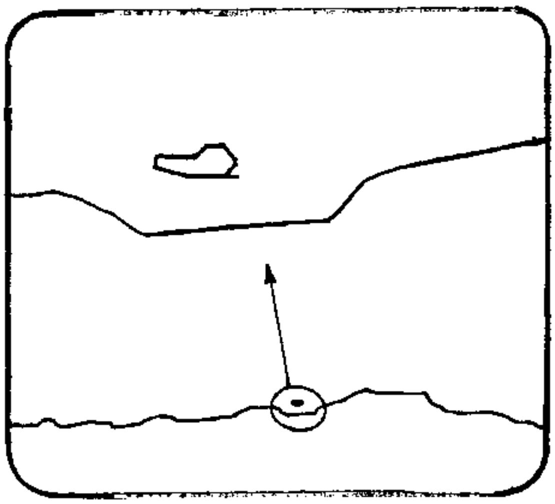


Fig. 7.3 Arrangement of the graphical simulation on the vector display terminal

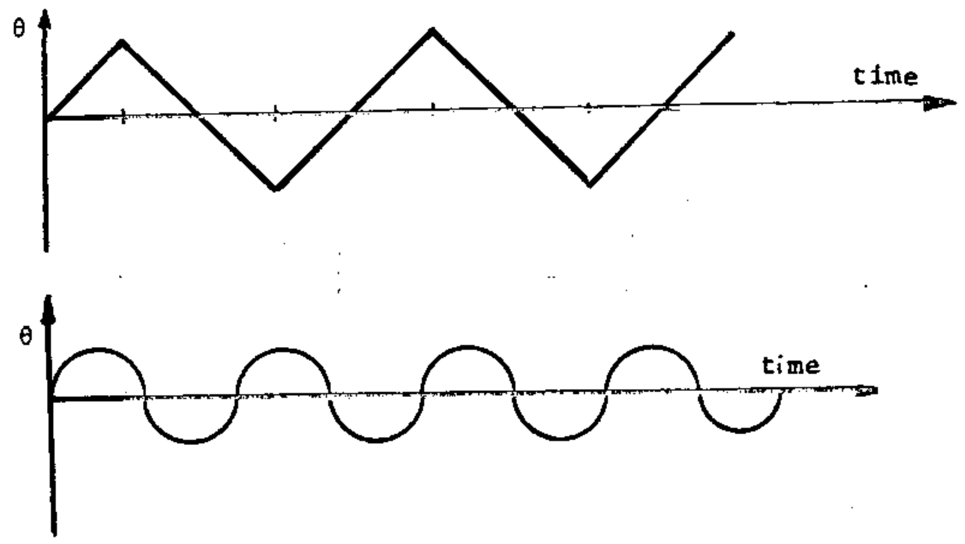
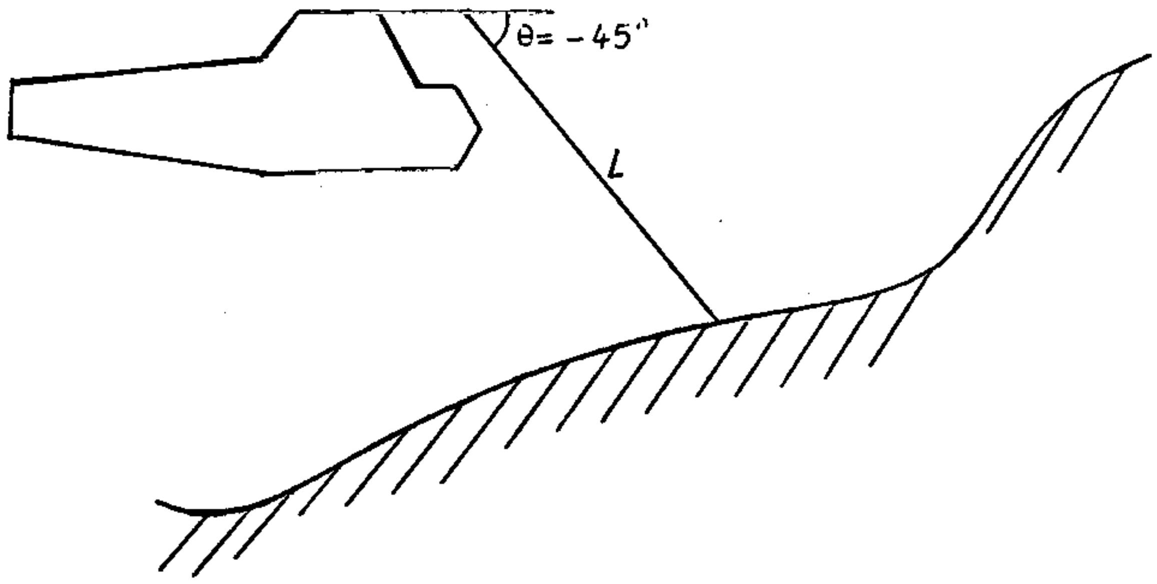


Fig. 7.4 Different motions of the first sonar.



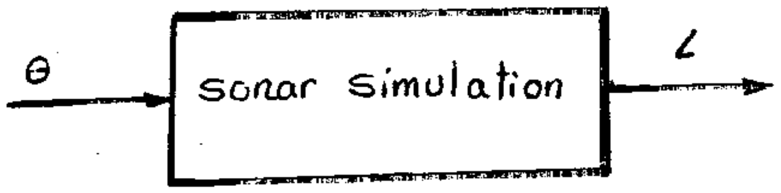
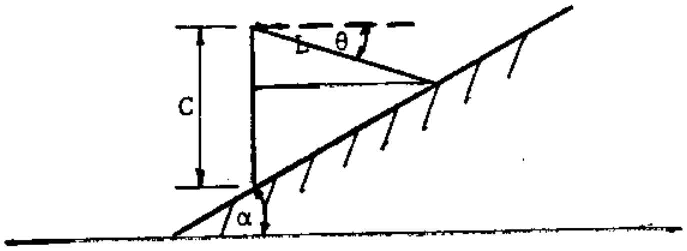


Fig. 7.5 Sonar simulation block diagram.



$$L*SIN(\theta)+L*COS(\theta)*TAN(\alpha)=C$$

$$SIN(\theta)+COS(\theta)*TAN(\alpha)=C/L$$

$$SIN(\theta)+B*COS(\theta)=A$$

Where:

$A=C/L$
$B=TAN(\alpha)$

SOLUTION FOR θ :

$$TAN(\theta)=\frac{B+A*SQRT(1+B**2-A**2)}{1-A**2}$$

Fig. 7.6 Calculation of the second sonar angle

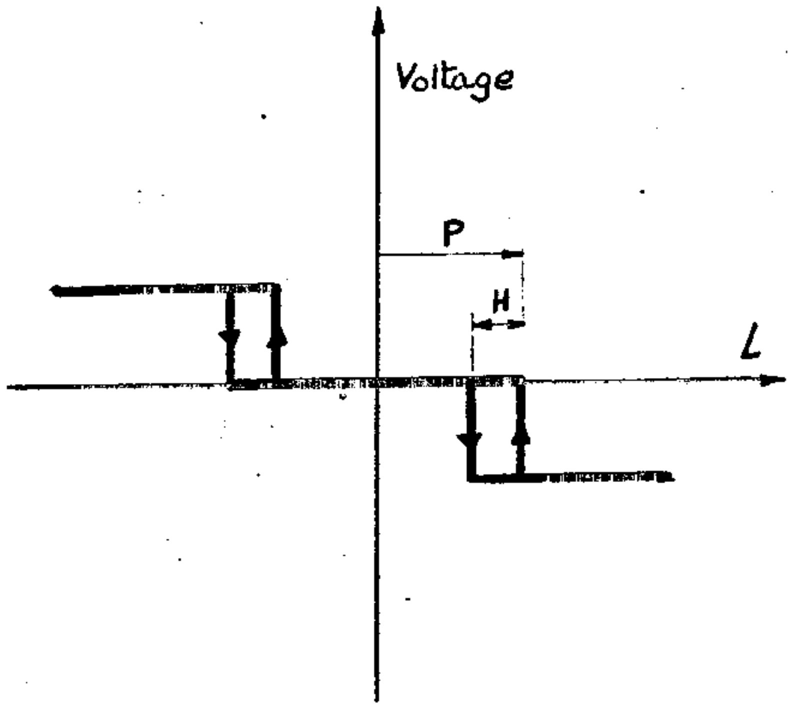


Fig. 7.7 Servocontroller for thrusters.

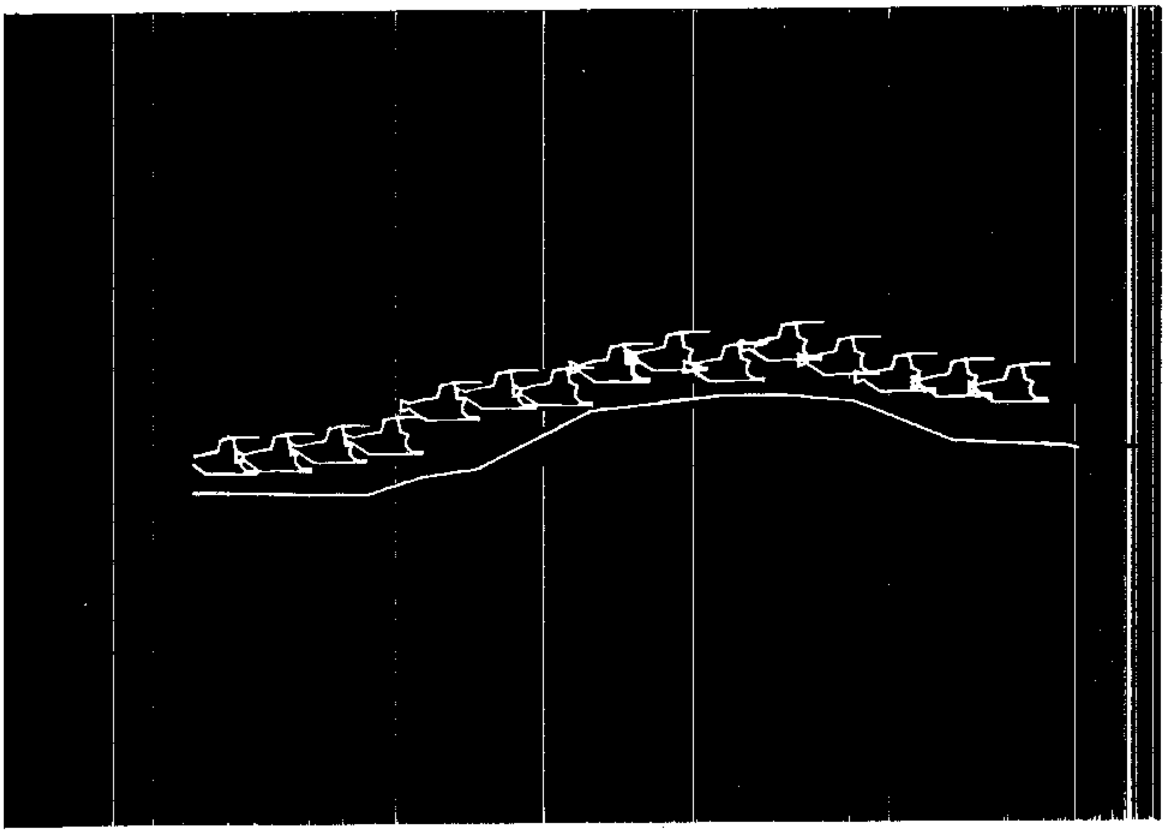


Fig. 7.8

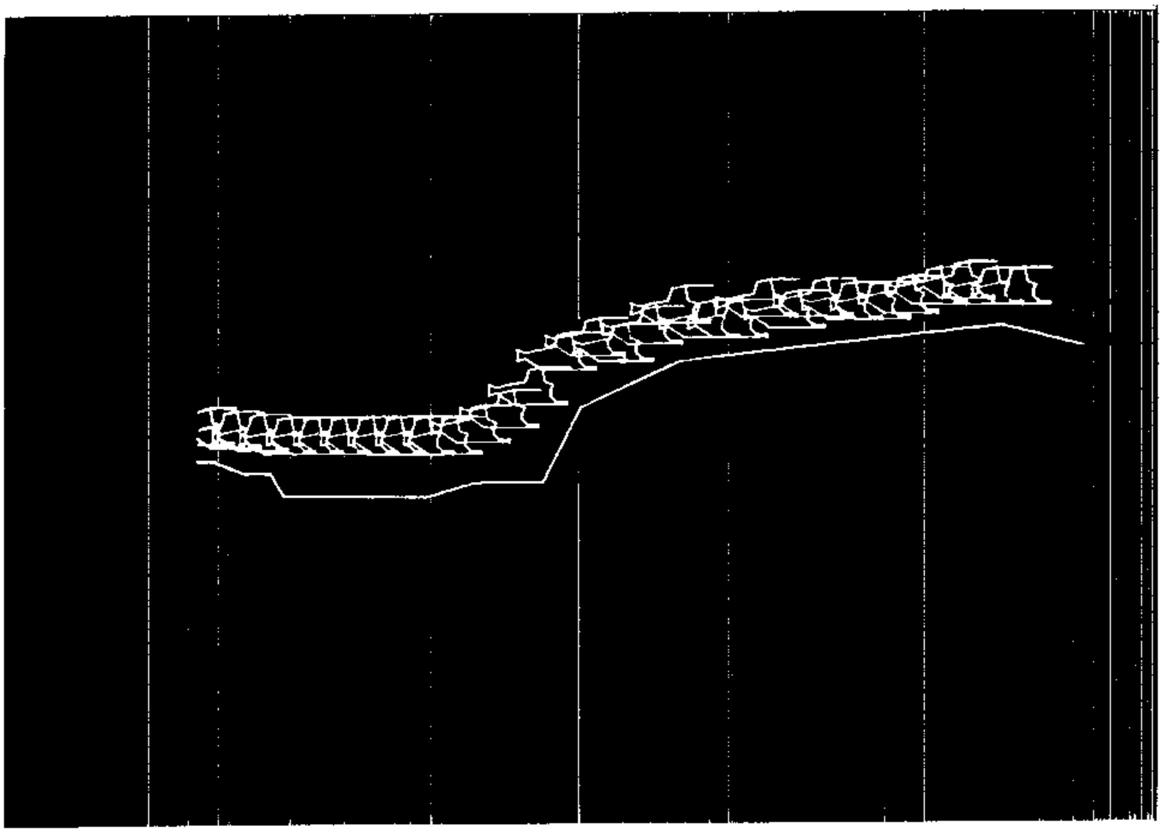


Fig. 7.9

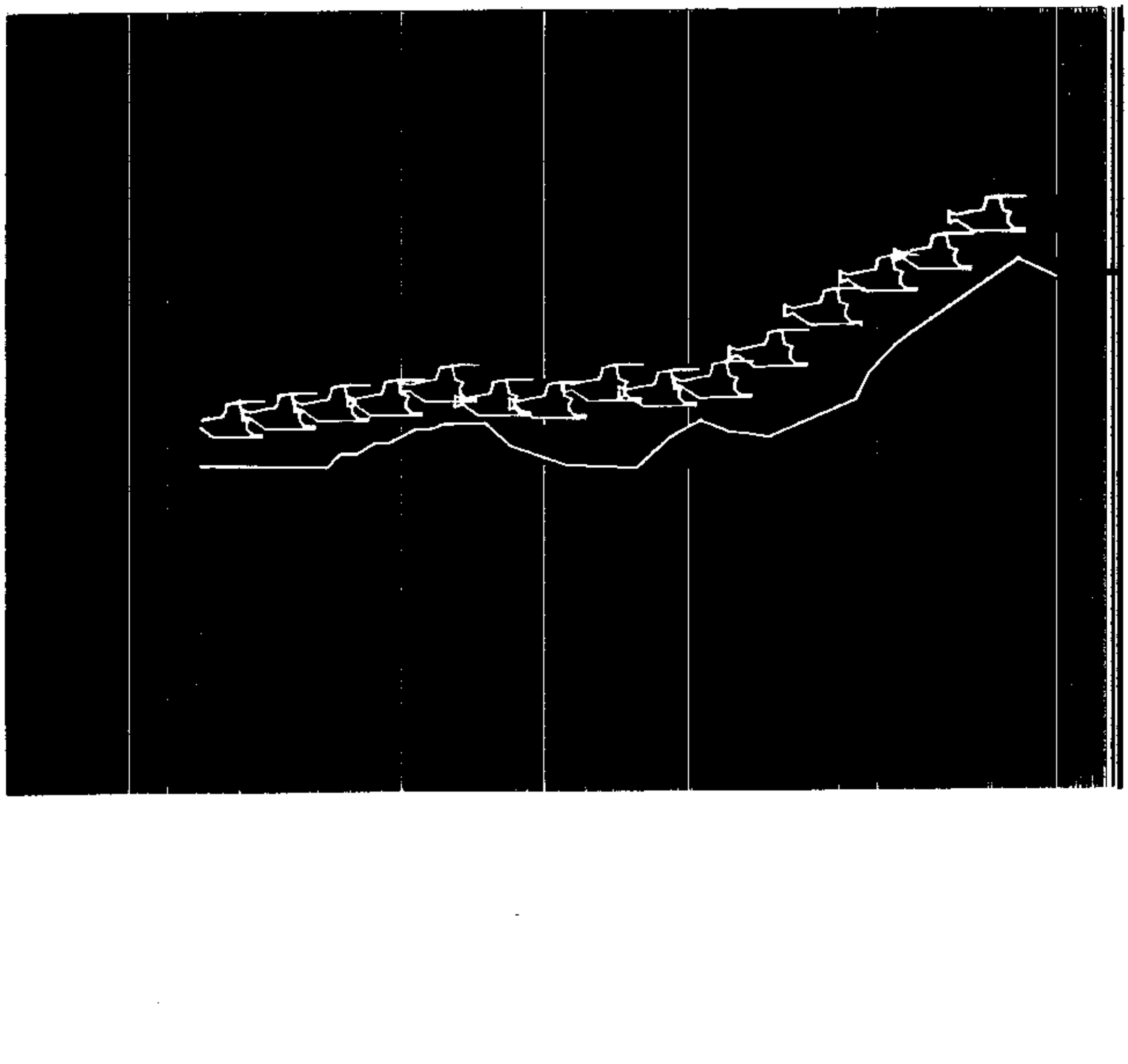


Fig. 7.10

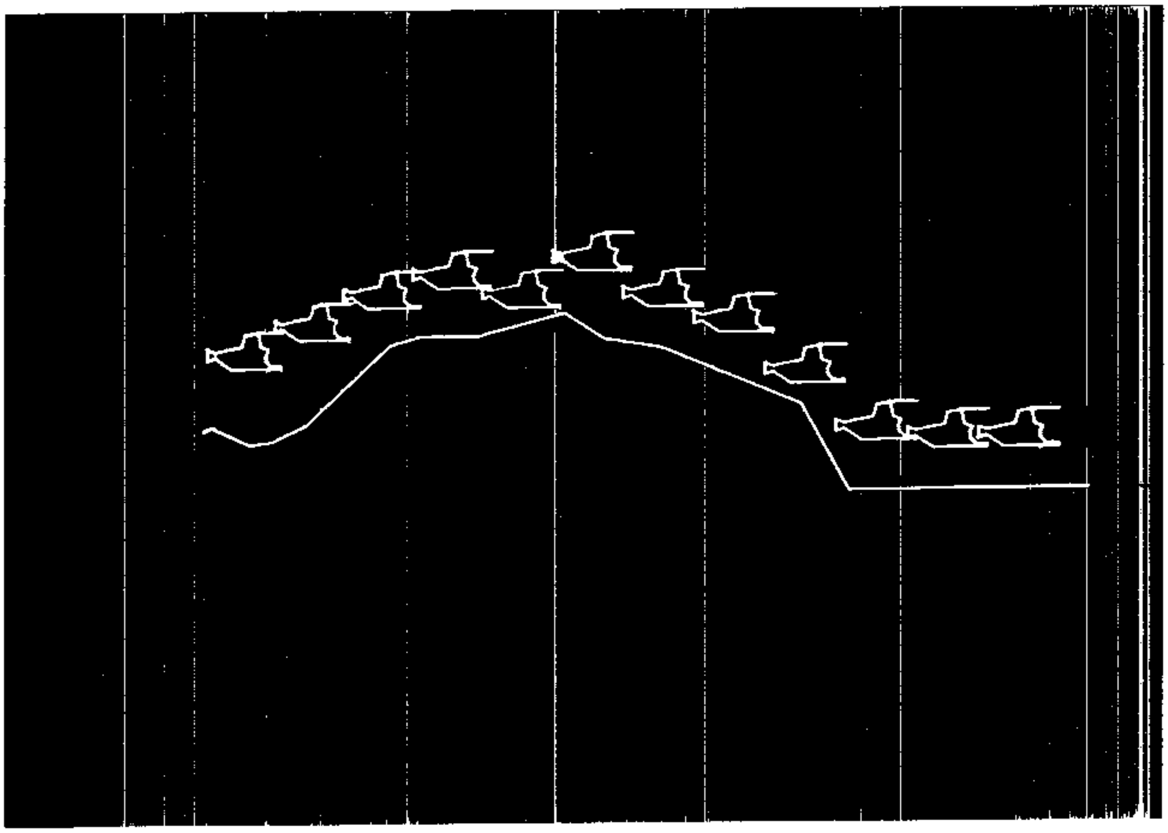


Fig. 7.11

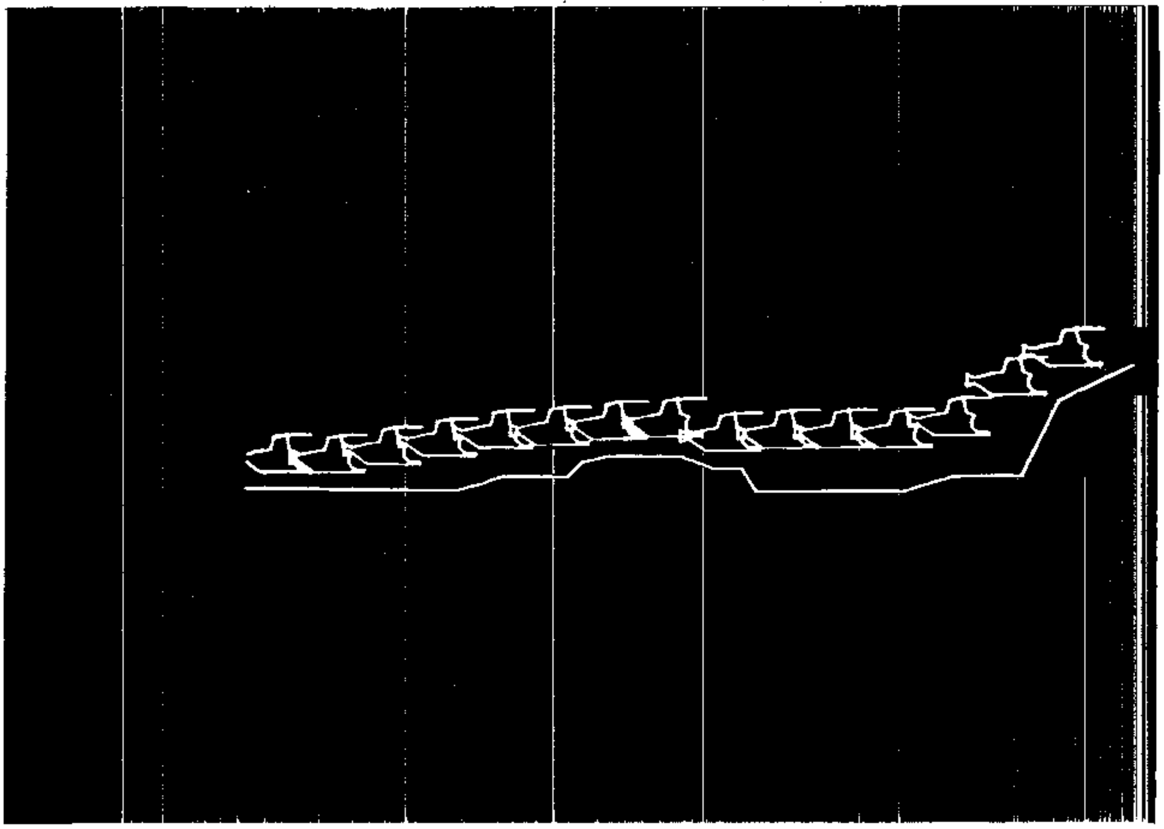


Fig. 7.12

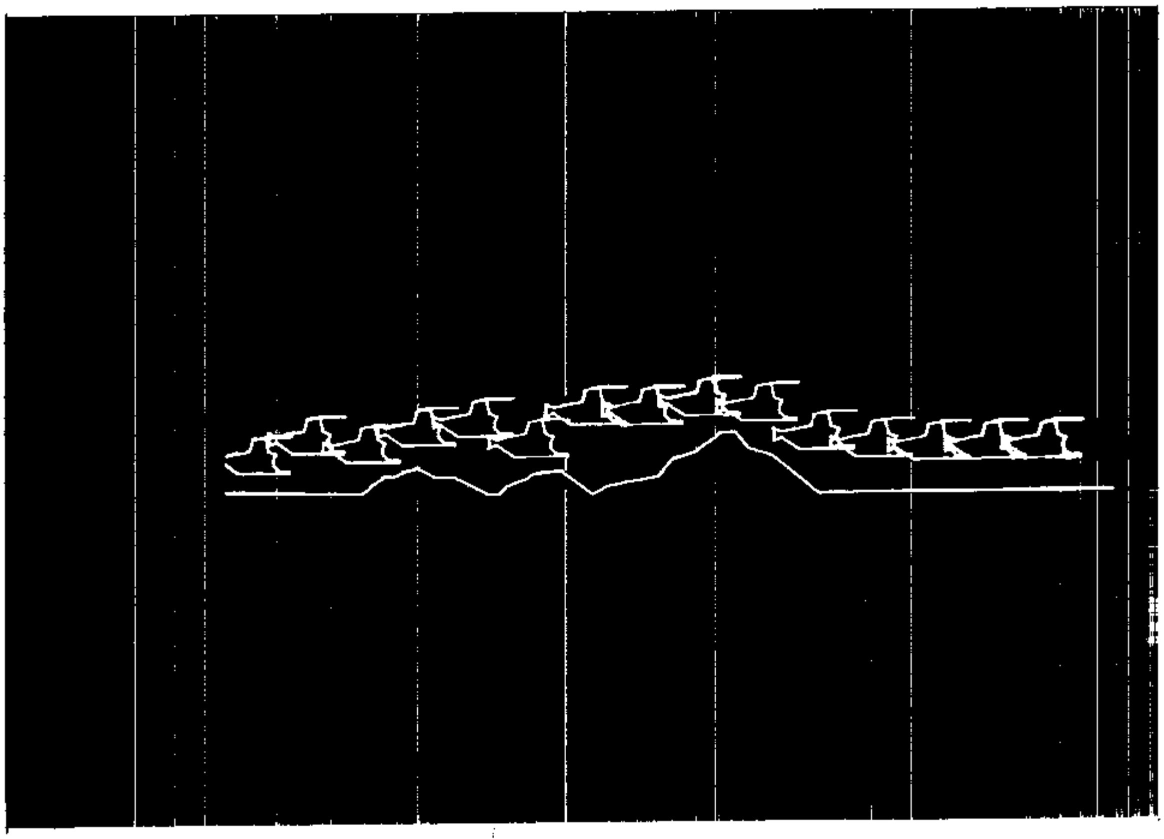


Fig. 7.13

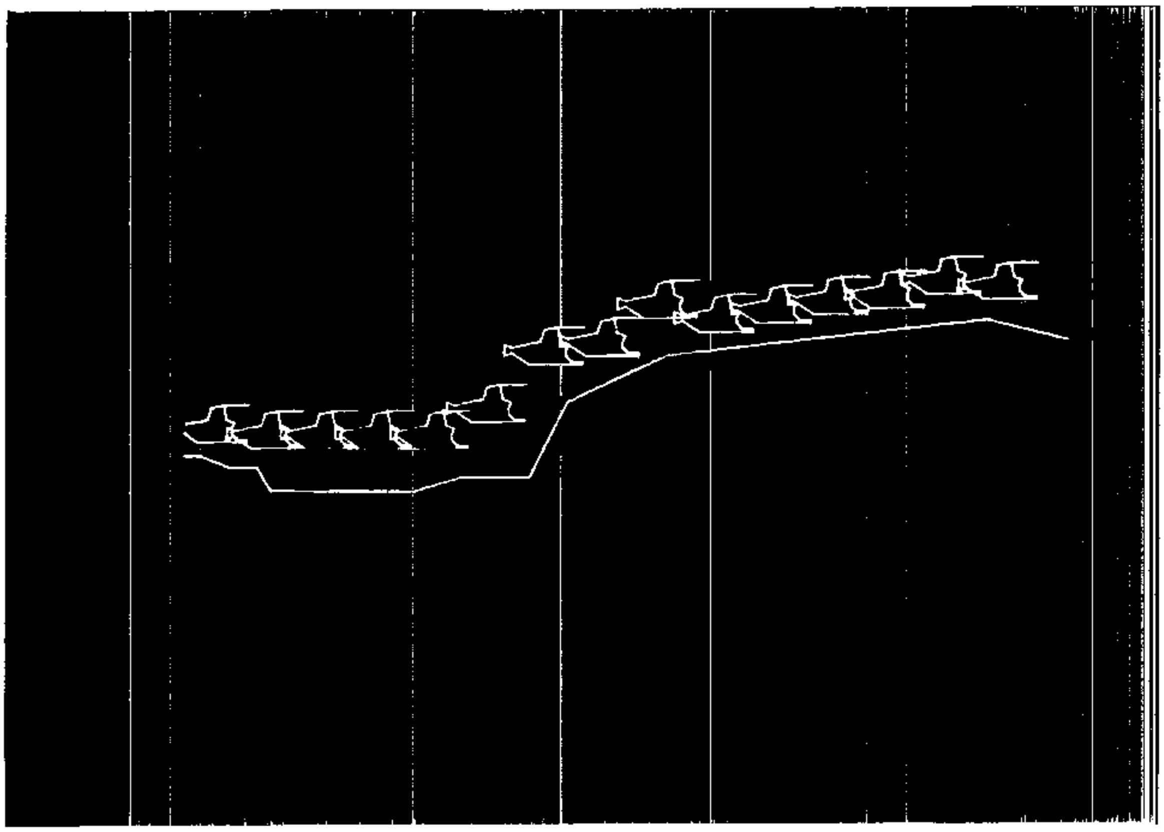


Fig. 7.14

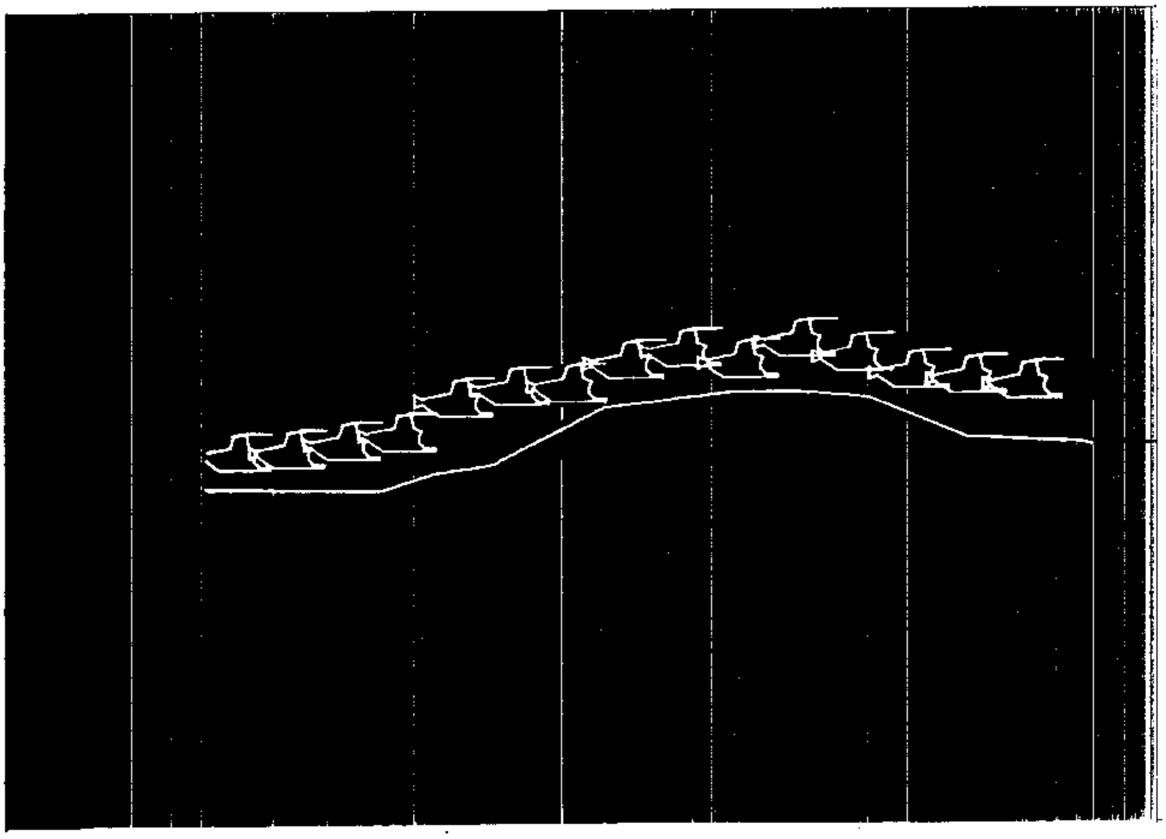


Fig. 7.15

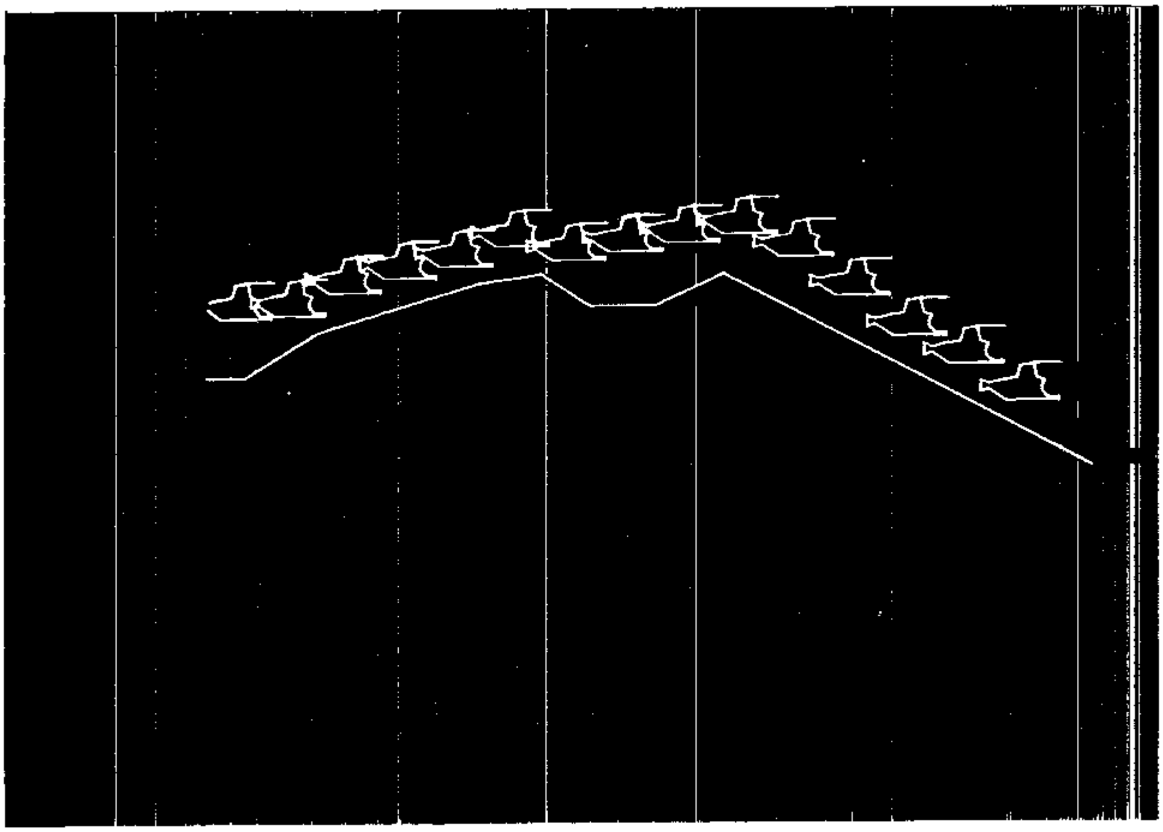


Fig. 7.16

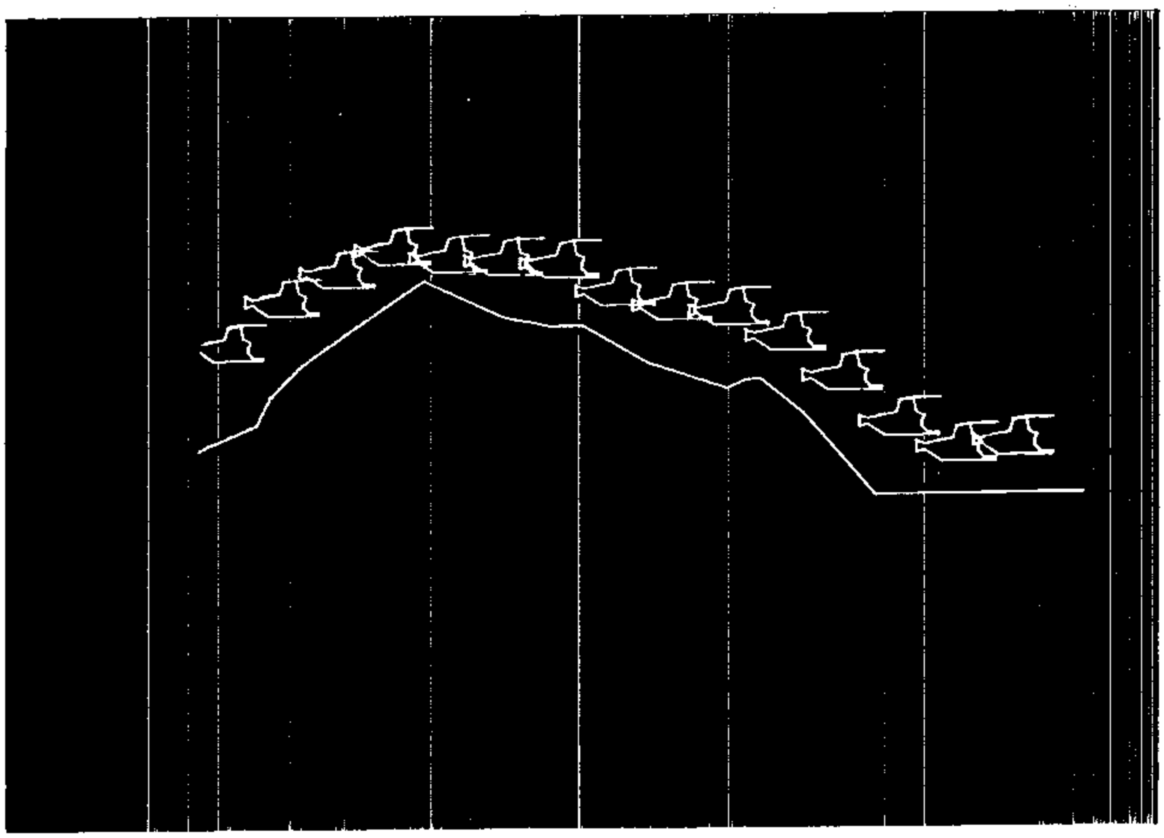


Fig. 7. 17

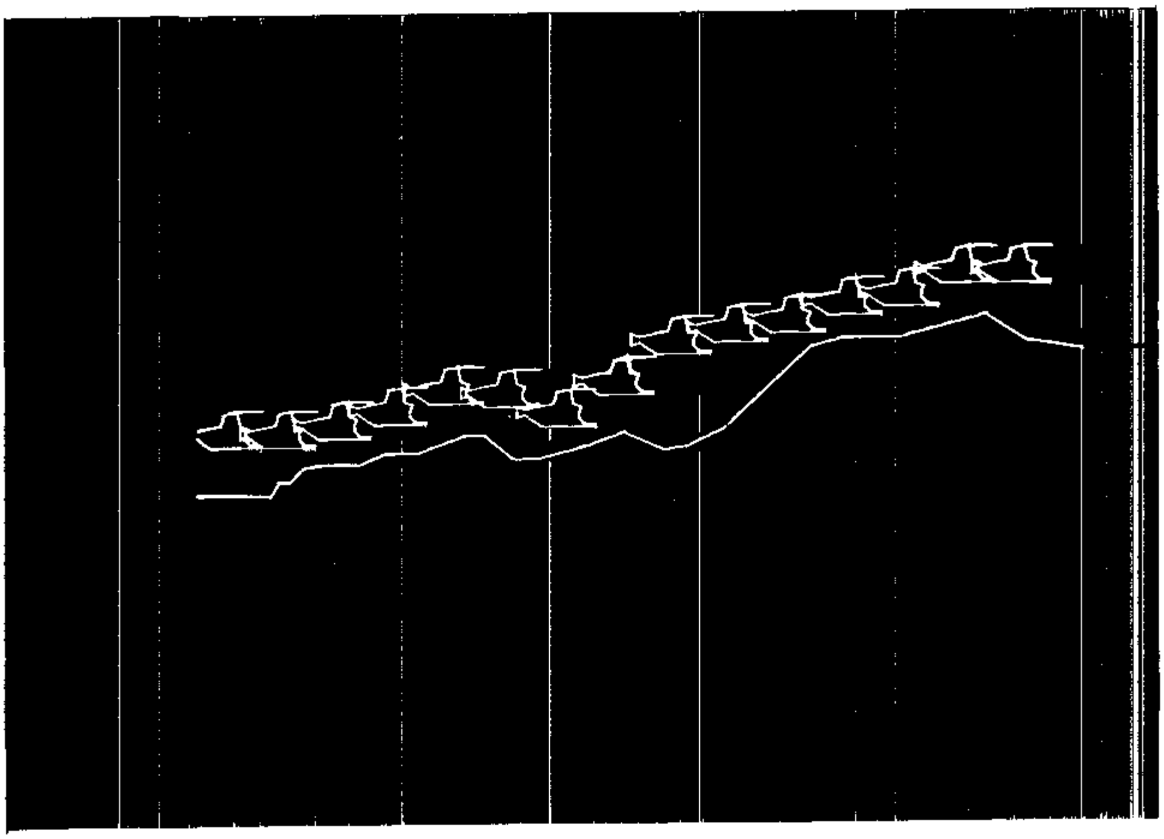


Fig. 7.18

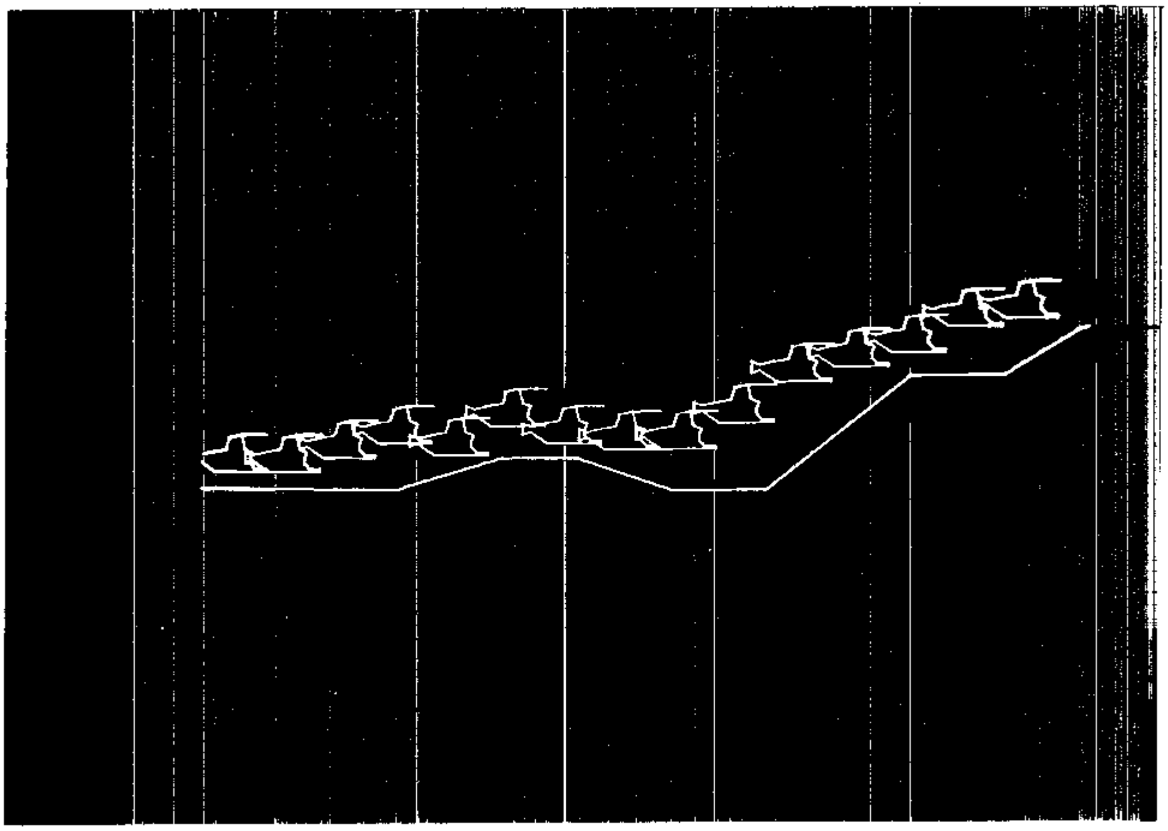


Fig. 7.19

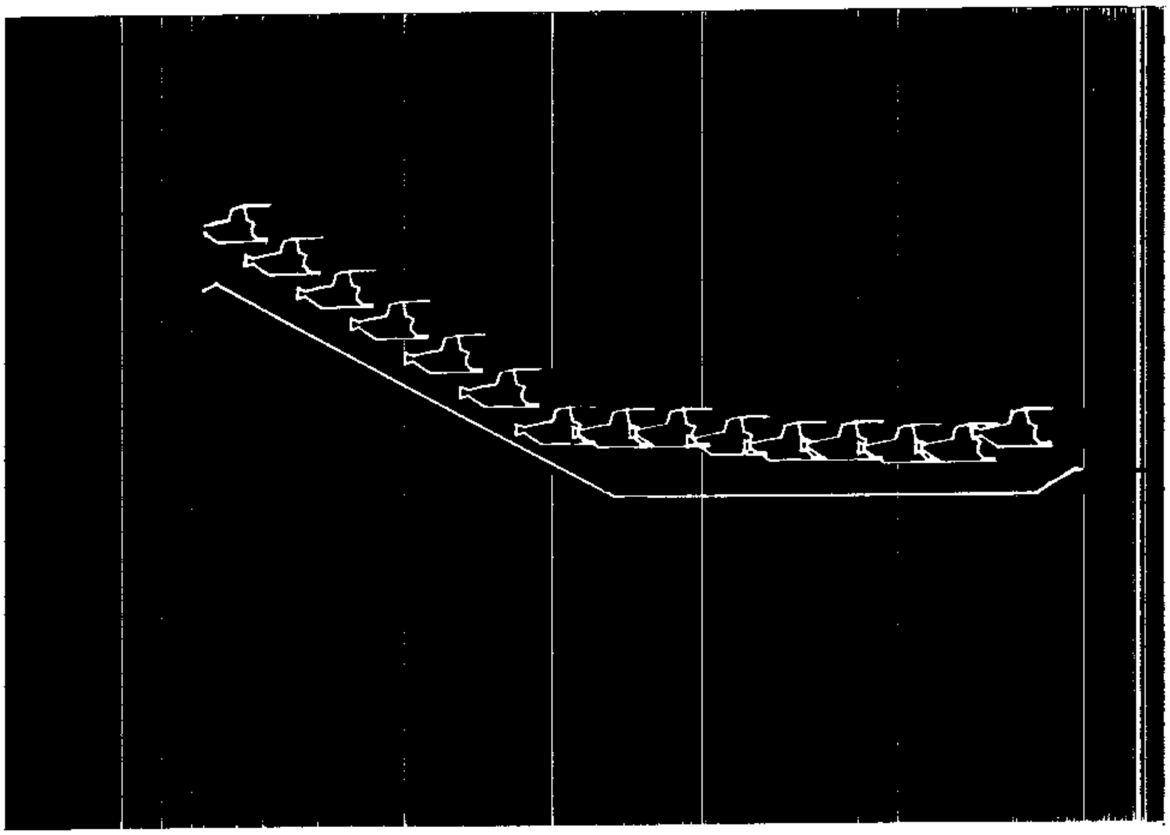


Fig. 7.20

CONCLUSIONS AND RECOMMENDATIONS

The purpose of this research was to simulate the motion and dynamics of underwater vehicles and to evaluate alternative control technology for following the ocean bottom. The equations described in chapter three have proven to be reliable and efficient to show the dynamics of underwater vehicles in real time. A more complete model can be used to simulate the motion of the vehicle ,but more terms in the differential equations would add to the complexity of the model, and consequently it would be hard to simulate thevehicle in real time on a PDP11/34.

The simulation is capable of characterizing the tight maneuvering and coupling among the motions of a wide range of underwater search vehicles. The coupling in different motions of the search vehicle is an important factor which was analyzed for controller design.

Looking back at chapters 4,5,6 and 7 suggest that the more control we have on the vehicle, the more complex tasks we would be able of perform. The controllers described in chapters 4,5,6 and 7 have shown good performance in orienting, positioning and bottom-following tasks.

Having the motion of the manipulator arm and the vehicle under the control of the operator will provide more capability of doing complex tasks; therefore future research should analyze the combined motion and dynamics of the manipulator arm and underwater vehicle.

8 GENERAL CHARACTERISTICS OF RCV-150 AND ALVIN

1) RCV-150

Hydro Products Model RCV-150 is a remote controlled, tethered vehicle equipped with television camera and lights, sonar, and a four degree of freedom manipulator. Propulsion is provided by four bidirectional, 10" diameter thrusters housed within Kort nozzles; two located at the aft end of the vehicle and two mounted at port and starboard on the upper plan of the amidships line. All four are fixed with respect to the vehicle. Hydraulic power (approx. 3 HP per motor) is provided to the thruster motors from an electrohydraulic converter which in turn receives electrical power through the tether cable. The television camera is mounted in the front of the vehicle on a pan and tilt mechanism. Control of the thrusters, camera, lights, and manipulator is accomplished with the aid of an on-board microcomputer and a second microcomputer on the surface. Communication between the two computers is through the tether cable.

The shape of the vehicle is roughly spherical with approximate overall dimensions of 58.5" long, 51.5" wide, and 44.5" high. Thruster positions are as indicated on the front and top views in Figure 2. Fore/aft motion is determined by the sum of the power delivered to the longitudinal thrusters,while heading (rotation about the z axis) is determined by the difference in the two thrusts. Likewise, vertical motion is produced by the sum of the upper two thrusters power and port/starboard motion is dependent on the difference in thrust.

The vehicle weighs approximately 1060 pounds in air and is about 15 pounds positively buoyant in water. Maximum speed forward is 2 knots and 1.5 knots in reverse. Transverse speed is limited to 1 knot and vertical motion is possible up to 1 knot. Thruster placement allows rotation about the z axis only and at rates of up to 90°/second. Rotation about the horizontal axes (pitch and roll) are stabilized by gravity. Since there is no active control about these axes, the television camera is designed to pan and tilt at rates of up to 60^{o}/second. Camera pan is limited to $\pm 45^{o}$ and tilt is restricted to $\pm 80^{o}$.

The tether between the launcher and vehicle is approximately 500 feet long and 0.6 inches in diameter and extends from the top of the vehicle. It is positively buoyant to lessen the possibility of ent anglement.

Buoyancy of the vehicle can be controlled only prior to launch with the addition or removal of ballast or syntactic foam.

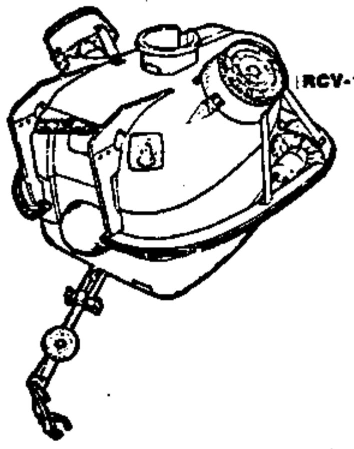
Modeling the vehicle as a sphere should give a fairly good estimate of the drag forces on the vehicle itself; however, any simulation of the vehicle should also include drag forces on the tether cable.

Unfortunately, some of the data required for developing a simulation (center of gravity, moments of inertia, center of buoyancy, drag forces) are not available at this time. The forward motion drag coefficient has been estimated as 1.5$\pm$ .5 from test pool thrust tests and the maximum forward speed. The vertical separation between the center of gravity and center of buoyancy ranges between 2 to 4 inches, depending on ballasting and optional equipment included.

RCV-150

SYSTEM CHARACTERISTICS

VEHICLE	
Standard Operating Depth	610 meters (2,000 feet)
Optional Maximum Operating Depth	2,000 meters (6,600 feet)
Maximum Operating Current	Greater than 2 knots
Viewing Range	4.5 meters (15 feet) clear water, no ambient light
Target Detection Range	30 feet to 20 inch dia. white target in clear water, no ambient light
Speed (still water at any operating depth)	Forward, 2 knots Reverse, 1.5 knots Port/Starboard, 1 knot Up/Down, 1 knot cw/ccw rotation, 90°/second
Direct Lift Capacity	40 kgs (88 pounds)
Emergency Location and Recovery Devices	Acoustic pinger, strobe flasher and tether cable cutter (standard)
Sensors, Standard	Television with internal pan and tilt Search sonar: sector sweep or side scan Vehicle pitch angle Vehicle roll angle Vehicle magnetic heading Vehicle turn rate Vehicle depth Altimeter Hydraulic oil temperature Seawater intrusion alarms Electronics temperature Hydraulic pressure


RCV-150 Vehicle

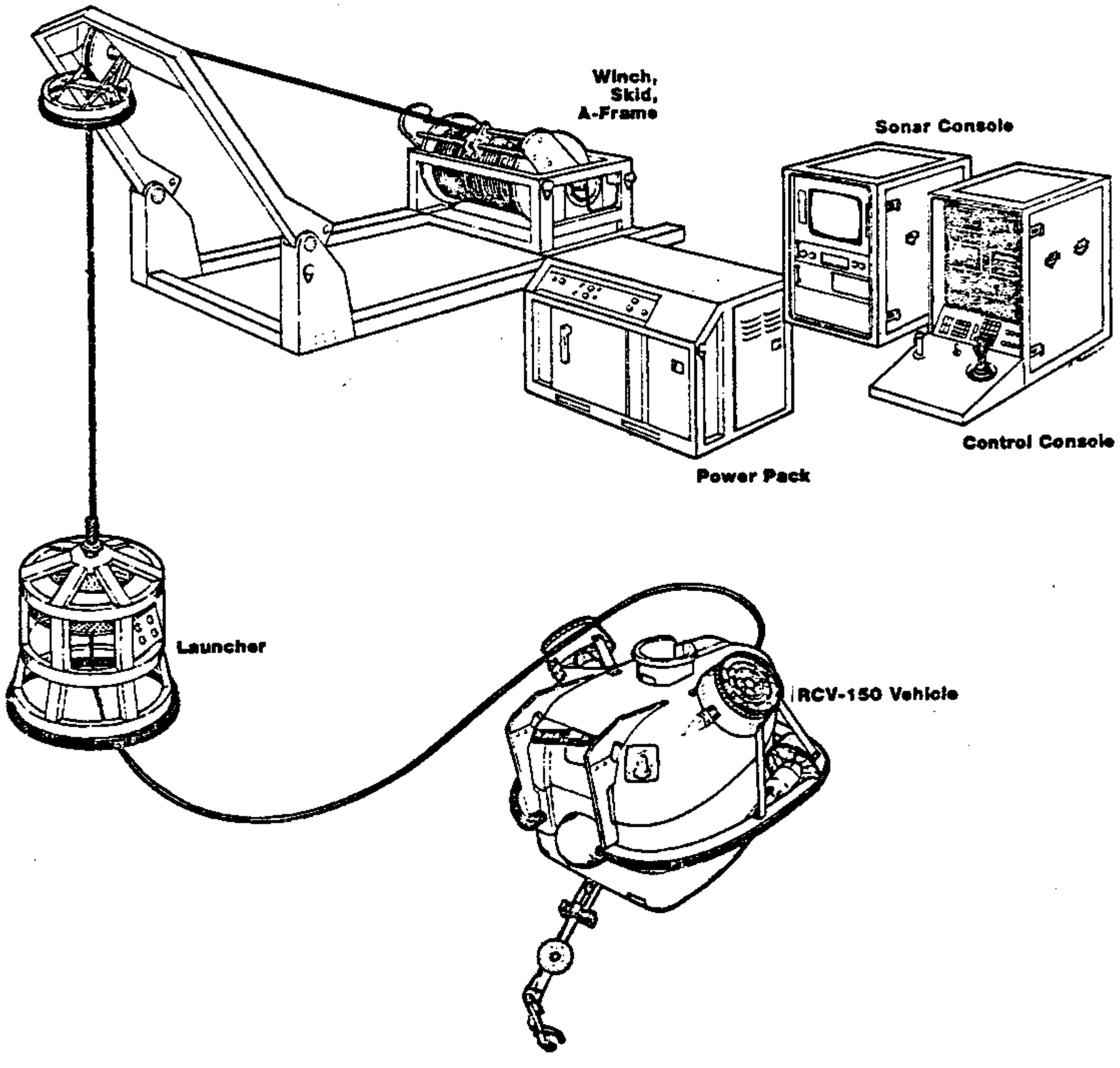
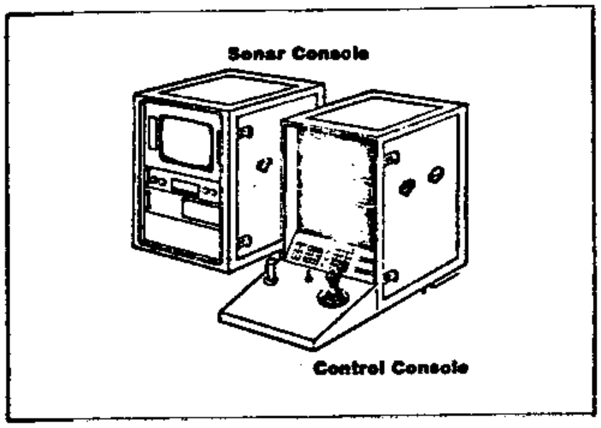


Figure 8.1 Major RCV-150 System Units

Sensors, Optional		Hydraulic, electrical power and control interfaces for customer requested options
Tools	Manipulator:	Four (4) degrees of freedom; total length 1 meter (3 feet); capable of handling 10 kg (22 pounds) of weight at full horizontal extension or 40 kg (88 pounds) directly below the vehicle
	Cable Cutter:	Abrasive cut-off saw; cuts 2 cm. (0.75 in.) dia. improved plow steel stranded cable in less than 3 minutes
	Soft Line Cutter:	Up to 2 cm. (0.75 in.) diameter polypropylene or manila line
Electronics Housing		Single housing contains telemetry, sensors, television camera, and micro computer control to minimize interconnect wiring between housings
Lights		Four (4) 250-watt tungsten quartz iodide lamps, individually controlled
Vehicle Power		15 electric hp (electro/hydraulic converter)
Propulsion		Four (4) 25 cm. (10 in.) diameter, thrusters: 2 longitudinal; 2 vertical/transverse (vertrans)
Vehicle Control		
Heading		Gravity stabilized
Depth		Automatic control, fully proportional
Fore & Aft, Port & Starboard		Automatic control, fully proportional
Pitch/Roll		Fully proportional in direction and speed
Structure		Noncorrosive, nonmagnetic, syntactic foam-filled glass reinforced plastic (GRP)
Buoyancy		Modular syntactic foam enclosed in fiberglass fairing

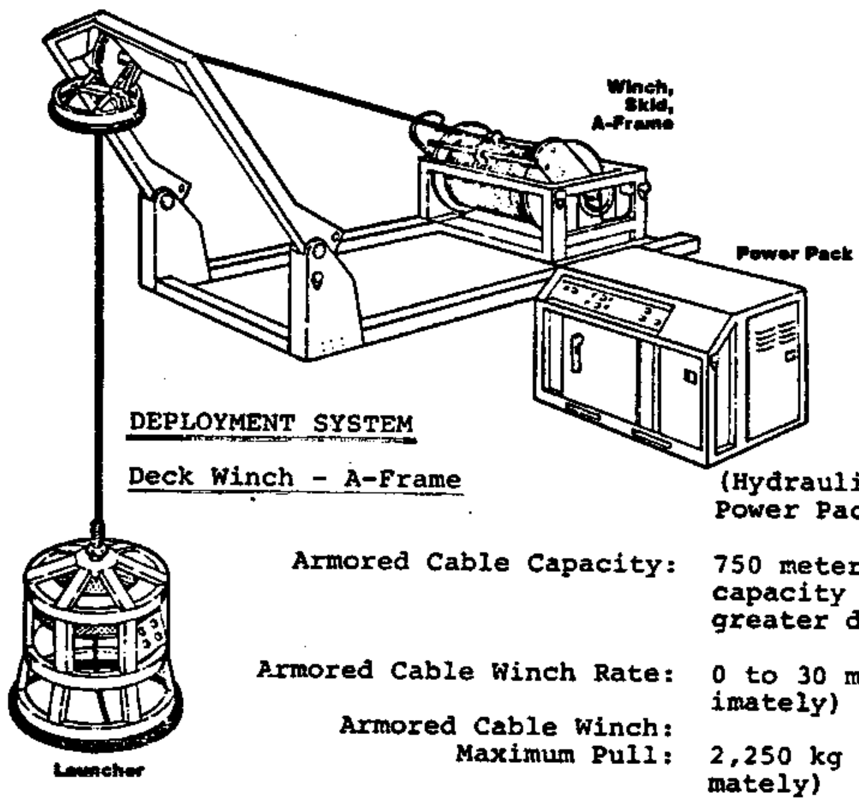


CONTROL CONSOLES

VEHICLE CONSOLE

Television Display Monitor	Video annotated with vehicle depth, heading, camera pan and tilt angles. Selectable second line with altitude, time of day, user ID
Data Display Monitor	Vehicle depth, altitude, magnetic heading, camera pan and tilt angles, vehicle pitch or roll angles, data error rate, time of day, number and direction of vehicle turns (tether cable twists), tether footage counter, inertial heading, temperature of electronics, user ID, hydraulic pressure, hydraulic oil temperature, launcher latch indicator, and alarm conditions
Auxiliary Display Monitor	Sonar display or optional display from customer supplied equipment
Bar Graph Displays	Vehicle thruster power levels, hydraulic pressure, electronics housing and hydraulic oil temperature, RF signal attenuation
LED Compass Rose	Vehicle heading

Illuminated Switch Bank	Vertical azimuth servo mode controls with reference choices of altimeter or depth and magnetic or inertial heading; data record; camera focus; light controls; tether cutter; cable cutter; stabilization; hydraulic isolation; and accessory controls
Alphanumeric Keyboard	Provides operator with ability to enter data into control and display systems and to access computer memory for system status
Vehicle Propulsion Joystick	Vehicle commands for proportional translation forward/reverse, left/right, up/down, and heading rotation cw/ccw; thruster trim control; and trigger and select switch to actuate manipulator jaws. Customer requested control options can be added
Camera Joystick	Positional control of television camera pan and tilt angles; automatic pan return to center
Manipulator Joystick	Provides positional control of manipulator assembly in 3 axes
Potentiometer Controls	Monitor brightness and contrast
Footpedal	Remote tether winch control
SONAR CONSOLE	High resolution sonar display monitor; sweep sector or side scan Video Processor--permits picture enhancement and display expansion. Sonar Control Unit -- permits range select, sector scan select, and transpond mode select capability



DEPLOYMENT SYSTEM

Deck Winch - A-Frame	(Hydraulically powered from Deck Power Pack)
Armored Cable Capacity:	750 meters (2,500 feet); higher capacity winches available for greater depths
Armored Cable Winch Rate:	0 to 30 m/min (100 ft/min approximately)
Armored Cable Winch: Maximum Pull:	2,250 kg (5,000 pounds approximately)
Underwater Launcher/Winch	
Tether Cable Capacity:	150 meters (500 feet) standard
Tether Cable Winch Rate:	25 m/min (80 ft/min approximately)
Tether Cable Winch Pull:	Approximately 23 kg (50 pounds up to 125 pounds adjustable)
Tether Cable	500 feet neutrally buoyant
Deck Power Pack	Control Panel: Main power on/off Vehicle power on/off Launcher in/out (tether) Deck winch power on/off Control station reset Main and vehicle voltage/current displays Includes 30 electric hp HPU to operate boom and deck winch Input power 220/440 vac; 3 phase; 50/60 hz 100 kva for start-up, 36 kva vehicle operating, 63 kva with winch and vehicle operating

Drag coefficient for translational motion along x-direction, Cx=55

$$F_x = Cx \cdot V_x^2$$

V_x=ft/sec
f=lbf
Cy=Cz=Cx

---

approximate moment of inertia around x-axis, Ix=2560 lb.ft

Iy=Iz=Ix

Drag coefficient for rotational motion C_ϕ=150

$$T = C_\phi \cdot P^2$$

P= radian/sec
T= lbf.ft
$C_\phi = C_\theta = C_\psi$

$Z_g = 3'$

RCV-150 SYSTEM DIMENSIONS

Item	Height	Width	Depth	Net Weight (Estimated)
Vehicle with EDO Sonar	1.1m (44.5 in.)	1.3m (51.5 in.)	1.5m (58.5 in.)	482 kg (1,060 lbs.)
Control Station	0.8m (31.0 in.)	0.6m (22.0 in.)	*	114 kg (250 lbs.)
Sonar Station	0.8m (31.0 in.)	0.6m (22.0 in.)	0.8m (32.0 in.)	68 kg (150 lbs.)
Winch/Skid/A-Frame with 2500 ft. Armored Cable	Without Launcher 2.9m (117.0 in.) Stored Position 3.0m (120.0 in.)	2.4m (96.0 in.)	4.5m (177.0 in.)	4,700 kg. (10,340 lbs.) 6,000 kg (13,200 lbs.) (including vehicle and launcher)
Power Pack	1.2m (46.0 in.)	1.5m (58.0 in.)	1.1m (42.0 in.)	818 kg (1,800 lbs.)
Underwater Launcher with 500 ft Tether Cable (Without Vehicle)	2.1m (83.0 in.	1.8m (72.0 in. dia.)	---	818 kg. (1,800 lbs.)

*With control shelf: 1.24m (49 in.)
With lid (shipping): 1.04m (41 in.)

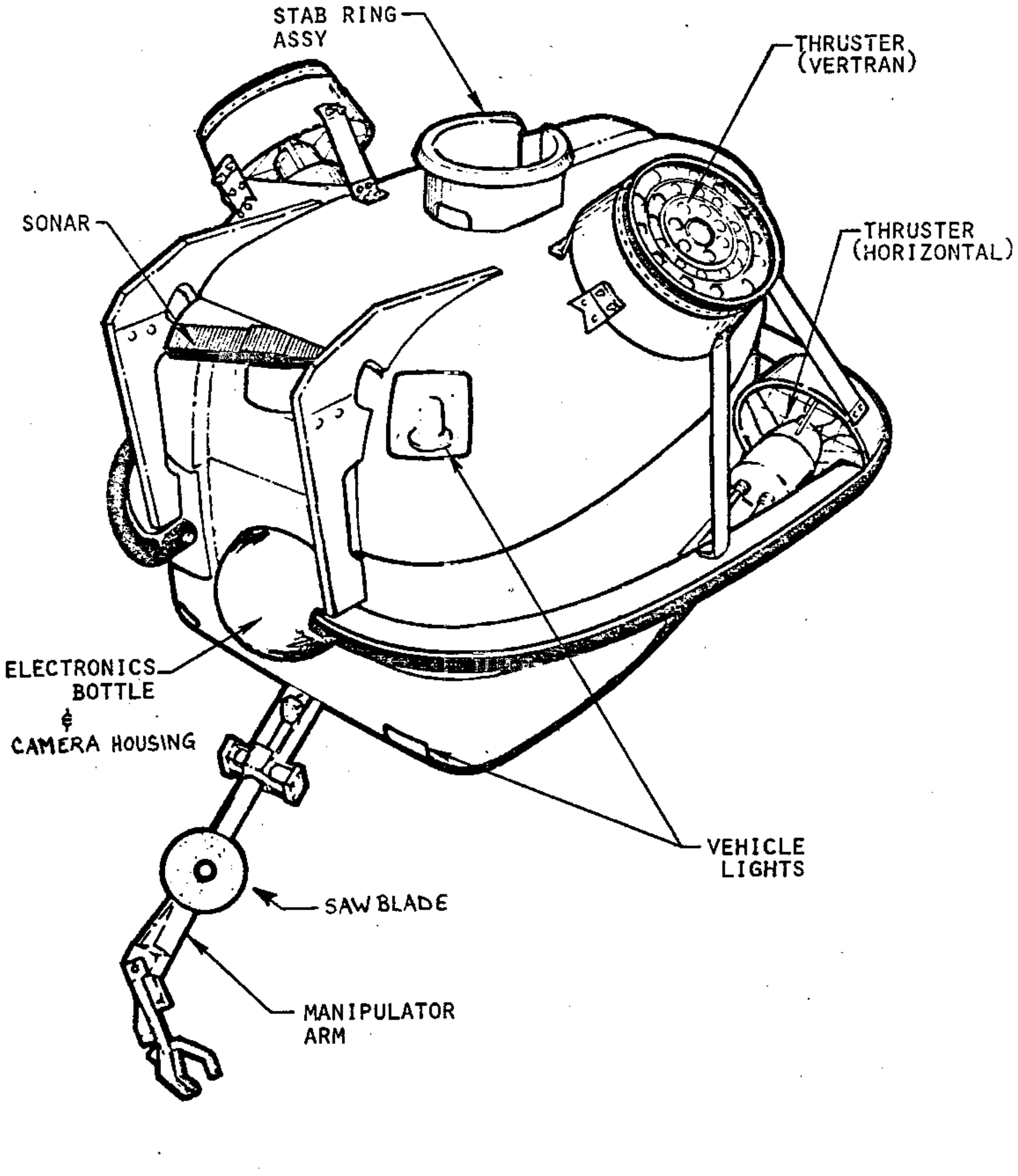


Figure 8.2 RCV-150 System Vehicle

2) ALVIN

In general, ALVIN's capabilities are as follows:

1. operate at any depth from the surface to 3658 meters at speeds of 0 - 3.5 Km/h
2. carry one or two observers and various internal and/or external instrumentation and tools
3. perform scientific or engineering tasks
4. maneuver within one foot of slopes or other bottom topography
5. rest on the bottom to perform tasks
6. hover at neutral bouyancy
7. remain submerged for periods up to approximately ten hours

The general characteristics of ALVIN are as follows:

length	- 7 meters (23 feet)
extreme beam	- 2.6 meters (8.5 feet)
air weight	- 14.65 tonnes (16.15 tons)
depth capability	- 3658 m (12,000 feet)
pay load	- up to 450 Kg (1000 lbs.)
personnel capacity	- 3
normal dive duration	- 6-10 hours
life support duration	- 210 man-hrs. (70 hrs./3 men)
speed	- 0-3.5 km/h (0-2 kts.)

Scale:	1-99.9 meters standard (other two and three digit scales available as an option).
Range:	33 meters typical. Maximum useable range will depend upon bottom type (rocks vs. mud) and other environmental characteristics.
Resolution:	0.1 meter.
Acoustic Level:	100 db,reference 1 bar at 1 yard (nominal).
Sample Rate:	2 Hz.
Supply Voltage:	15 to 35 VDC.
Supply Current:	20 ma nominal.
Operating Frequency:	100 kHz.
Depth Rating:	12,000 meters.
Temperature Rating:	-2 C to 50 C (28 F to 120 F).
Length:	27.3 cm (10.75 inches).
Diameter:	12.4 cm (4.9 inches).

1. dry weight 34600 lb.

2. cruising speed forward 1.2 mph

   vertical 1.0 mph

3. emergency speed forward 2.2 mph

   vertical 1.8 mph

4. center of gravity (Figure 2)

5. radius of gyration 2.0 ft.

6. drag coefficients: forward .20

   up .33

   down .26

7. thrust of stern prop 80 lb.

8. thrust of each control prop 65 lb.

9. time constant of stern prop 1 sec.

10. time constant of stern prop direction control motor 0.2 sec.

11. rate of direction change of stern prop 10 deg/sec.

12. maximum direction change +25 deg.

13. time constant of control props 0.3 sec.

14. rate of direction change of control props.

    right prop 26 degrees/sec.
    left prop 30 degrees/sec.

For more technical information about ALVIN
please refer to Erik Vaaler's thesis, 1982.

REFERENCES

1. Sheridan, Thomas B., MIT, personal communication, 1982.

2. Takahashi, M., "Design of an Experimental Simulation For a Human Remote Control of an Undersea Vehicle", Master's Thesis MIT, August 1979.

3. Tani, K., "Supervisory Control of Remote Manipulation With Compensation For Moving Target", Man-Machine Systems Laboratory Report, MIT, 1980.

4. Winey C. M., "Computer Simulated Visual and Tactile Feedback as an Aid to Manipulator and Vehicle Control", Master's Thesis MIT, May 1981.

5. Abkowitz A. M., "Stability and Control of Ocean Vehicles", MIT Press, Cambridge, MA., 1969.

6. Bishop R.E.D., "Mechanics of Marine Vehicles" ,University of London, England, 1975 .

7. Welty J., "Momentum Transfer " John Wiley, 1976 .

8. Franklin G. E., "Digital Control", Addison-Wesley Publishing Co., 1980 .

9. Dorf R.C.,"Modern Control Systems", Addison-Welsey Publishing Co., 1980 .

```
      SUBROUTINE SUBM8
      COMMON/DMABUF/IDUM(3894),TETAB,TUNAB,TETAFB,TUNAFB,HT,VT,ICOM,
     1 TNGT
      COMMON/NAME2/XX,YY,ALEU,ALED,CDIS,VAL,ILI,UHN,UVN,XE,YE,TUNA
     *,XEF,YEF,TUNAF
      IXRD=-15
      IYRD=-6
      IXR=2*IXRD
      IYR=2*IYRD
      IXE=XE
      IYE=YE
      IXEF=XEF
      IYEF=YEF
      IXX=XX
      IYY=YY
      IF(INI.EQ.1) GO TO 10
C     ALEU=THE LENGTH OF THE ENVIRONMENT THAT APPEARS ON THE UPPER
C     PART OF THE SCREEN.
C     ALED=THE LENGTH OF THE ENVIRONMENT THAT APPEARS ON THE LOWER
C     PART OF THE SCREEN.
C
C     THE MAXIMUM RATIO OF THE ALEU TO ALED IS 16.
C
C
      NT=1
      ALEU=ALEU*2
      TYPE *,'1 OR -1'
      ACCEPT *,NAD
      CALL MGINIT
      CALL SETINT(13)
      SCU=4000/ALEU
      INT1=0
      INT2=0
      INT3=0
      INT4=0
      G1=-1500./SCU
      G2=1000.+G1
      G3=G2
      G4=1000.+G2
      G5=-2000.
      G6=-1000.+G1
      G7=G4
      G8=2000
      G9=G6
      G10=G1
      G11=-G1
      G22=1000.+G1
      G33=G22-3000.
      G44=G22-2000.
      SHIFT=500.
      HI=-1000.*SCU
      SCD=4000./ALED
      SCALEU=ALEU/8000
      SCALED=ALED/4000
C
C     IN THIS SECTION THE LOWER ENVIRONMENT DATA IS READ IN FOUR SECTIONS.
C
      OPEN(UNIT=1,NAME='PATH.DAT',TYPE='OLD')
      CALL NPOINT(IL1)
      CALL SETINT(0)
      CALL NPOINT(IENL1)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
1000  READ(1,*)IDN,IXC,IYC
      IF(IDN.EQ.1)CALL MOVEI(IXC,IYC)
      IF(IDN.EQ.2)CALL DRAWI(IXC,IYC)
      IF(IDN.EQ.0) GO TO 2000
      GO TO 1000
2000  CALL MGSEND
      CALL NPOINT(IL2)
      CALL SETINT(0)
      CALL NPOINT(IENL2)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
3000  READ(1,*)IDN,IXC,IYC
      IF(IDN.EQ.1)CALL MOVEI(IXC,IYC)
      IF(IDN.EQ.2)CALL DRAWI(IXC,IYC)
      IF(IDN.EQ.0)GO TO 4000
      GO TO 3000
4000  CALL MGSEND
      CALL NPOINT(IL3)
      CALL SETINT(0)
      CALL NPOINT(IENL3)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
5000  READ(1,*)IDN,IXC,IYC
      IF(IDN.EQ.1)CALL MOVEI(IXC,IYC)
      IF(IDN.EQ.2)CALL DRAWI(IXC,IYC)
      IF(IDN.EQ.0)GO TO 6000
      GO TO 5000
6000  CALL MGSEND
      CALL NPOINT(IL4)
      CALL SETINT(0)
      CALL NPOINT(IENL4)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
7000  READ(1,*)IDN,IXC,IYC
      IF(IDN.EQ.1)CALL MOVEI(IXC,IYC)
      IF(IDN.EQ.2)CALL DRAWI(IXC,IYC)
      IF(IDN.EQ.0)GO TO 8000
      GO TO 7000
8000  CALL MGSEND
      CLOSE(UNIT=1,DISPOSE='SAVE')
C
C
C     IN THIS SECTION THE LOWER ENVIRONMENT DATA IS READ IN FOUR SECTIONS.
C
      OPEN(UNIT=1,NAME='PATH.DAT',TYPE='OLD')
      CALL NPOINT(IL1R)
      CALL SETINT(0)
      CALL NPOINT(IENL1R)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
71    READ(1,*)IDN,IXC,IYC
      IF(IDN.EQ.1)CALL MOVEI(IXC,IYC)
      IF(IDN.EQ.2)CALL DRAWI(IXC,IYC)
      IF(IDN.EQ.0) GO TO 72
      GO TO 71
72    CALL MGSEND
      CALL NPOINT(IL2R)
      CALL SETINT(0)
      CALL NPOINT(IENL2R)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
73    READ(1,*)IDN,IXC,IYC
      IF(IDN.EQ.1)CALL MOVEI(IXC,IYC)
      IF(IDN.EQ.2)CALL DRAWI(IXC,IYC)
      IF(IDN.EQ.0)GO TO 74
      GO TO 73
74    CALL MGSEND
      CALL NPOINT(IL3R)
      CALL SETINT(0)
      CALL NPOINT(IENL3R)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
75    READ(1,*)IDN,IXC,IYC
      IF(IDN.EQ.1)CALL MOVEI(IXC,IYC)
      IF(IDN.EQ.2)CALL DRAWI(IXC,IYC)
      IF(IDN.EQ.0)GO TO 76
      GO TO 75
76    CALL MGSEND
      CALL NPOINT(IL4R)
      CALL SETINT(0)
      CALL NPOINT(IENL4R)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
77    READ(1,*)IDN,IXC,IYC
      IF(IDN.EQ.1)CALL MOVEI(IXC,IYC)
      IF(IDN.EQ.2)CALL DRAWI(IXC,IYC)
      IF(IDN.EQ.0)GO TO 78
      GO TO 77
78    CALL MGSEND
      CLOSE(UNIT=1,DISPOSE='SAVE')
C
C
C     IN THIS SECTION THE LOWER ENVIRONMENT DATA IS READ IN FOUR SECTIONS.
C
      OPEN(UNIT=1,NAME='PATH.DAT',TYPE='OLD')
      CALL NPOINT(IL1L)
      CALL SETINT(0)
      CALL NPOINT(IENL1L)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
171   READ(1,*)IDN,IXC,IYC
      IF(IDN.EQ.1)CALL MOVEI(IXC,IYC)
      IF(IDN.EQ.2)CALL DRAWI(IXC,IYC)
      IF(IDN.EQ.0) GO TO 172
      GO TO 171
172   CALL MGSEND
      CALL NPOINT(IL2L)
      CALL SETINT(0)
      CALL NPOINT(IENL2L)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
173   READ(1,*)IDN,IXC,IYC
      IF(IDN.EQ.1)CALL MOVEI(IXC,IYC)
      IF(IDN.EQ.2)CALL DRAWI(IXC,IYC)
      IF(IDN.EQ.0)GO TO 174
      GO TO 173
174   CALL MGSEND
      CALL NPOINT(IL3L)
      CALL SETINT(0)
      CALL NPOINT(IENL3L)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
175   READ(1,*)IDN,IXC,IYC
      IF(IDN.EQ.1)CALL MOVEI(IXC,IYC)
      IF(IDN.EQ.2)CALL DRAWI(IXC,IYC)
      IF(IDN.EQ.0)GO TO 176
      GO TO 175
176   CALL MGSEND
      CALL NPOINT(IL4L)
      CALL SETINT(0)
      CALL NPOINT(IENL4L)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
177   READ(1,*)IDN,IXC,IYC
      IF(IDN.EQ.1)CALL MOVEI(IXC,IYC)
      IF(IDN.EQ.2)CALL DRAWI(IXC,IYC)
      IF(IDN.EQ.0)GO TO 178
      GO TO 177
178   CALL MGSEND
      CLOSE(UNIT=1,DISPOSE='SAVE')
C
C
C     THE DATA FOR UPPER ENVIRONMENT IS READ.
C
C
      OPEN(UNIT=1,NAME='PATH.DAT',TYPE='OLD')
      CALL NPOINT(ILI1)
      CALL SETINT(0)
      CALL NPOINT(IENM1)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
101   READ(1,*)IDN,IXC,IYC
      IYC=IYC+SHIFT
      IF(IDN.EQ.1)CALL MOVEI(2*IXC,2*IYC)
      IF(IDN.EQ.2)CALL DRAWI(2*IXC,2*IYC)
      IF(IDN.EQ.0) GO TO 102
      GO TO 101
102   CALL MGSEND
      CALL NPOINT(ILI2)
      CALL SETINT(0)
      CALL NPOINT(IENM2)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
```

```
103     READ(1,*)IDN,IXC,IYC
        IYC=IYC+SHIFT
        IF(IDN.EQ.1)CALL MOVEI(2*IXC,2*IYC)
        IF(IDN.EQ.2)CALL DRAWI(2*IXC,2*IYC)
        IF(IDN.EQ.0)GO TO 104
        GO TO 103
104     CALL MGSEND
        CALL NPOINT(ILI3)
        CALL SETINT(0)
        CALL NPOINT(IENM3)
        CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
105     READ(1,*)IDN,IXC,IYC
        IYC=IYC+SHIFT
        IF(IDN.EQ.1)CALL MOVEI(2*IXC,2*IYC)
        IF(IDN.EQ.2)CALL DRAWI(2*IXC,2*IYC)
        IF(IDN.EQ.0)GO TO 106
        GO TO 105
106     CALL MGSEND
        CALL NPOINT(ILI4)
        CALL SETINT(0)
        CALL NPOINT(IENM4)
        CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
107     READ(1,*)IDN,IXC,IYC
        IYC=IYC+SHIFT
        IF(IDN.EQ.1)CALL MOVEI(2*IXC,2*IYC)
        IF(IDN.EQ.2)CALL DRAWI(2*IXC,2*IYC)
        IF(IDN.EQ.0)GO TO 108
        GO TO 107
108     CALL MGSEND
        CLOSE(UNIT=1,DISPOSE='SAVE')
        CALL NPOINT(IC)
        CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
        CALL SETINT(13)
        CALL MOVEI(18+IXR,IYR)
        CALL DRAWI(12+IXR,2+IYR)
        CALL DRAWI(10+IXR,12+IYR)
        CALL DRAWI(IXR,10+IYR)
        CALL DRAWI(-2+IXR,2+IYR)
        CALL DRAWI(-22+IXR,-2+IYR)
        CALL DRAWI(-28+IXR,IYR)
        CALL DRAWI(-28+IXR,-8+IYR)
        CALL DRAWI(-22+IXR,-6+IYR)
        CALL DRAWI(-10+IXR,-14+IYR)
        CALL DRAWI(28+IXR,-14+IYR)
        CALL DRAWI(28+IXR,-12+IYR)
        CALL DRAWI(20+IXR,-12+IYR)
        CALL DRAWI(16+IXR,-8+IYR)
        CALL DRAWI(16+IXR,-4+IYR)
        CALL DRAWI(18+IXR,IYR)
        CALL MOVEI(10+IXR,12+IYR)
        CALL DRAWI(30+IXR,12+IYR)
        CALL MGSEND
        CALL NPOINT(I1)
        CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
        CALL NPOINT(I2)
        CALL SETINT(10)
        CALL MOVEI(-28+IXR,-4+IYR)
        CALL DRAWI(-48+IXR,-4+IYR)
        CALL DRAWI(-40+IXR,-2+IYR)
        CALL MOVEI(-48+IXR,-4+IYR)
        CALL DRAWI(-40+IXR,-6+IYR)
        CALL MGSEND
        CALL NPOINT(I3)
        CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
        CALL NPOINT(I4)
        CALL SETINT(10)
        CALL MOVEI(18+IXR,-4+IYR)
        CALL DRAWI(38+IXR,-4+IYR)
        CALL DRAWI(30+IXR,-2+IYR)
        CALL MOVEI(38+IXR,-4+IYR)
        CALL DRAWI(30+IXR,-6+IYR)
```

```
      CALL MGSEND
      CALL NPOINT(I5)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
      CALL NPOINT(I6)
      CALL SETINT(10)
      CALL MOVEI(IXR,10+IYR)
      CALL DRAWI(IXR,30+IYR)
      CALL DRAWI(2+IXR,22+IYR)
      CALL MOVEI(IXR,30+IYR)
      CALL DRAWI(-2+IXR,22+IYR)
      CALL MGSEND
      CALL NPOINT(I7)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
      CALL NPOINT(I8)
      CALL SETINT(10)
      CALL MOVEI(IXR,-14+IYR)
      CALL DRAWI(IXR,-34+IYR)
      CALL DRAWI(2+IXR,-26+IYR)
      CALL MOVEI(IXR,-34+IYR)
      CALL DRAWI(-2+IXR,-26+IYR)
      CALL MGSEND
      CALL NPOINT(ID)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
      CALL SETINT(10)
      CALL MOVEI(9+IXRD,IYRD)
      CALL DRAWI(6+IXRD,1+IYRD)
      CALL DRAWI(5+IXRD,6+IYRD)
      CALL DRAWI(IXRD,5+IYRD)
      CALL DRAWI(-1+IXRD,1+IYRD)
      CALL DRAWI(-11+IXRD,-1+IYRD)
      CALL DRAWI(-14+IXRD,IYRD)
      CALL DRAWI(-14+IXRD,-4+IYRD)
      CALL DRAWI(-11+IXRD,-3+IYRD)
      CALL DRAWI(-5+IXRD,-7+IYRD)
      CALL DRAWI(14+IXRD,-7+IYRD)
      CALL DRAWI(14+IXRD,-6+IYRD)
      CALL DRAWI(10+IXRD,-6+IYRD)
      CALL DRAWI(8+IXRD,-4+IYRD)
      CALL DRAWI(8+IXRD,-2+IYRD)
      CALL DRAWI(9+IXRD,IYRD)
      CALL MOVEI(5+IXRD,6+IYRD)
      CALL DRAWI(15+IXRD,6+IYRD)
      CALL MGSEND
C
C
C
      CALL SETINT(9)
      CALL NPOINT(IDUMY)
      CALL LDTRN3(1,0.,0.,0.,0.,1.,0.,0.)
      CALL NPOINT(IRADD1)
      CALL MOVEI(0,0)
      CALL NPOINT(IRADD2)
      CALL DRAWI(0,0)
      CALL MGSEND
C
C
C
      CALL SETINT(9)
      CALL NPOINT(IDUMF)
      CALL LDTRN3(1,0.,0.,0.,0.,1.,0.,0.)
      CALL NPOINT(IRAFF1)
      CALL MOVEI(0,0)
      CALL NPOINT(IRAFF2)
      CALL DRAWI(0,0)
      CALL MGSEND
C
C
C
C
      CALL SETINT(9)
      CALL NPOINT(IDUMY1)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
      CALL NPOINT(IRAD1)
      CALL MOVEI(0,0)
      CALL NPOINT(IRAD2)
      CALL DRAWI(0,0)
      CALL MGSEND
C
C
C
      CALL SETINT(9)
      CALL NPOINT(IDUMF1)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
      CALL NPOINT(IRAF1)
      CALL MOVEI(0,0)
      CALL NPOINT(IRAF2)
      CALL DRAWI(0,0)
      CALL MGSEND
C
C
C
C     HERE WE DRAW A VEHICLE RIGTH ON THE UPPER PLANE OF THE SCREEN.
C     THE VEHICLE DOES NOT MOVE. THE ENVIROMENT OF THE VEHICLE MOVES.
      IF(VAL.NE.0)GO TO 24
      CALL MODIFY(IC)
      CALL LDTRN3(SCU,0.,0.,0.,0.,SCU,0.,-400.)
C
C
C
C     HERE WE DRAW TEH ENVIROMENT ON THE LOWER PART OF THE SCREEN.
C     THE ENVIROMENT DOES NOT MOVE.
24    CALL MODIFY(IL1)
      CALL SETINT(10)
      CALL MODIFY(IENL1)
      CALL LDTRN3(SCD,0.,0.,CDIS-2000.*SCD,0.,SCD,0.,-1800.)
      CALL MODIFY(IL2)
      CALL SETINT(10)
      CALL MODIFY(IENL2)
      CALL LDTRN3(SCD,0.,0.,CDIS-1000.*SCD,0.,SCD,0.,-1800.)
      CALL MODIFY(IL3)
      CALL SETINT(10)
      CALL MODIFY(IENL3)
      CALL LDTRN3(SCD,0.,0.,CDIS,0.,SCD,0.,-1800.)
      CALL MODIFY(IL4)
      CALL SETINT(10)
      CALL MODIFY(IENL4)
      CALL LDTRN3(SCD,0.,0.,CDIS+1000.*SCD,0.,SCD,0.,-1800.)
      CALL MGSEND
      CALL MODIFY(IL3L)
      CALL SETINT(10)
      CALL MODIFY(IENL3L)
      CALL LDTRN3(SCD,0.,0.,CDIS-4000.*SCD,0.,SCD,0.,-1800.)
      CALL MODIFY(IL4L)
      CALL SETINT(10)
      CALL MODIFY(IENL4L)
      CALL LDTRN3(SCD,0.,0.,CDIS-3000.*SCD,0.,SCD,0.,-1800.)
      CALL MGSEND
      CALL MODIFY(IL1R)
      CALL SETINT(10)
      CALL MODIFY(IENL1R)
      CALL LDTRN3(SCD,0.,0.,CDIS+2000.*SCD,0.,SCD,0.,-1800.)
      CALL MODIFY(IL2R)
      CALL SETINT(10)
      CALL MODIFY(IENL2R)
      CALL LDTRN3(SCD,0.,0.,CDIS+3000.*SCD,0.,SCD,0.,-1800.)
      CALL MGSEND
C
C
C
      INI=1
10    XU=XX/SCALEU
      XD=XX/SCALED
```

```
         YU=YY/SCALEU
         YD=YY/SCALED
C
3        IF(XD.GE.0) GO TO 4
         IF(XD.LT.0) GO TO 5
C
C        IN LOWER ENVIROMENT ONLY THE VEHICLE MOVES
C
C
C
4        XRD=AMOD((XD+2000.*SCD),4000.*SCD)-2000.*SCD
         CRY=AMOD((XX+2000.),4000.)-2000.
         GO TO 6
5        XRD=AMOD((XD-2000.*SCD),-4000.*SCD)+2000.*SCD
         CRY=AMOD((XX-2000.),-4000.)+2000.
C
C
C
6        XRDD=XRD+CDIS
         IF(VAL.NE.0)GO TO 25
         YYU=-YU+100.
         YYD=YD-1500.
         GO TO 26
25       YYU=100.
         YYD=-1500.+YD
         CALL MODIFY(IC)
         CALL LDTRN3(SCU,0.,0.,0.,0.,SCU,0.,-400.+YU)
26       IF(NT.EQ.1)GO TO 777
         NT=1
         GO TO 999
777      NT=0
999      IB=10*NT
         IN2=(UHN-60)/120+1
         IN4=-(UHN-60)/120
         IN8=(UVN-60)/120+1
         IN6=-(UVN-60)/120
         IB2=IN2*IB
         IB4=IN4*IB
         IB6=IN6*IB
         IB8=IN8*IB
         CALL MODIFY(I2)
         CALL SETINT(IB2)
         CALL MODIFY(I4)
         CALL SETINT(IB4)
         CALL MODIFY(I6)
         CALL SETINT(IB6)
         CALL MODIFY(I8)
         CALL SETINT(IB8)
         YUDM=YU*VAL
         CALL MODIFY(I1)
         CALL LDTRN3(SCU,0.,0.,0.,0.,SCU,0.,-400.+YUDM)
         CALL MODIFY(I3)
         CALL LDTRN3(SCU,0.,0.,0.,0.,SCU,0.,-400.+YUDM)
         CALL MODIFY(I5)
         CALL LDTRN3(SCU,0.,0.,0.,0.,SCU,0.,-400.+YUDM)
         CALL MODIFY(I7)
         CALL LDTRN3(SCU,0.,0.,0.,0.,SCU,0.,-400.+YUDM)
C
C
C
         IF(CRY.LE.G2.AND.CRY.GE.G1)GO TO 500
         IF(CRY.LT.G4.AND.CRY.GT.G3)GO TO 501
         IF(CRY.LT.G6.AND.CRY.GE.G5)GO TO 502
         IF(CRY.LE.G8.AND.CRY.GE.G7)GO TO 504
         IF(CRY.LE.G10.AND.CRY.GE.G9)GO TO 503
500      DIS1=0
         DIS2=(-2*CRY-2000)*SCU
         DIS3=-2*CRY*SCU
         DIS4=0
         INT1=0
         INT2=13
         INT3=13
```

```
      INT4=0
      SCU1=0
      SCU2=SCU
      SCU3=SCU
      SCU4=0
      IF(DIS3.GT.-2010.)GO TO 13
      DIS2=0
      INT2=0
      SCU2=0
      GO TO 13
501   DIS1=0
      DIS2=0
      DIS3=-2*CRY*SCU
      DIS4=(-2*CRY+2000)*SCU
      INT1=0
      INT2=0
      INT3=13
      INT4=13
      SCU1=0
      SCU2=0
      SCU3=SCU
      SCU4=SCU
      IF(DIS4.GT.-2010.)GO TO 13
      INT3=0
      DIS3=0
      SCU3=0
      GO TO 13
502   DIS1=(-2*CRY-4000)*SCU
      DIS2=0
      DIS3=0
      DIS4=(-2*CRY-6000)*SCU
      INT1=13
      INT2=0
      INT3=0
      INT4=13
      SCU1=SCU
      SCU2=0
      SCU3=0
      SCU4=SCU
      IF(DIS1.GT.-2010.)GO TO 13
      INT4=0
      DIS4=0
      SCU4=0
      GO TO 13
503   DIS2=(-2*CRY-2000)*SCU
      DIS1=(-2*CRY-4000)*SCU
      DIS3=0
      DIS4=0
      INT1=13
      INT2=13
      INT3=0
      INT4=0
      SCU1=SCU
      SCU2=SCU
      SCU3=0
      SCU4=0
      IF(DIS2.GT.-2010)GO TO 13
      INT1=0
      DIS1=0
      SCU1=0
      GO TO 13
504   DIS1=(-2*CRY+4000)*SCU
      DIS2=0
      DIS3=0
      DIS4=(-2*CRY+2000)*SCU
      INT1=13
      INT2=0
      INT3=0
      INT4=13
      SCU1=SCU
      SCU2=0
      SCU3=0
      SCU4=SCU
13    CALL MODIFY(ILI1)
      CALL SETINT(INT1)
      CALL MODIFY(ILI2)
      CALL SETINT(INT2)
      CALL MODIFY(ILI3)
      CALL SETINT(INT3)
      CALL MODIFY(ILI4)
      CALL SETINT(INT4)
      CALL MODIFY(IENM1)
      THIS=YYU+HI
      CALL LDTRN3(SCU1,0.,0.,DIS1,0.,SCU1,0.,THIS-500.)
      CALL MODIFY(IENM2)
      CALL LDTRN3(SCU2,0.,0.,DIS2,0.,SCU2,0.,THIS-500.)
      CALL MODIFY(IENM3)
      CALL LDTRN3(SCU3,0.,0.,DIS3,0.,SCU3,0.,THIS-500.)
      CALL MODIFY(IENM4)
      CALL LDTRN3(SCU4,0.,0.,DIS4,0.,SCU4,0.,THIS-500.)
      CALL MODIFY(ID)
      CALL LDTRN3(SCD,0.,0.,XRDD,0.,SCD,0.,YYD-300.)
      CALL MODIFY(IDUMY)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
      IXRDD=XRDD
      IYYD=YYD
C
C
C
      CALL MODIFY(IRADD1)
      CALL MOVEI(IXRDD,IYYD-300)
      IF(IXE.GE.0)GO TO 44
      IF(IXE.LT.0)GO TO 45
44    IXEL=AMOD((IXE+2000.*SCD),4000.*SCD)-2000.*SCD
      GO TO 46
45    IXEL=AMOD((IXE-2000.*SCD),-4000.*SCD)+2000.*SCD
46    IGS=IXEL*SCD+CDIS
      IG=IYE*SCD-1500
      IF (TUNA.EQ.10000)GO TO 212
      IF (IGS.GT.2000)GO TO 212
      GO TO 211
212   IGS=IXRDD
      IG=IYYD
211   CALL MODIFY(IRADD2)
      CALL DRAWI(IGS,IG-300)
C
C
      IIYU=YU*VAL+100.
      IIXU=0.
      CALL MODIFY(IDUMY1)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
      CALL MODIFY(IRAD1)
      CALL MOVEI(IIXU,IIYU-500)
      DIMF=ABS((XE-XX)/SCALEU)
      IF(DIMF.GT.30000.)GO TO 213
      IXF=(XE-XX)/SCALEU
      IYF=(YE)/SCALEU+100-YU*(1-VAL)
      IF (TUNA.EQ.10000)GO TO 213
      IF(IXF.GT.2000)GO TO 213
      GO TO 214
213   IXF=IIXU
      IYF=IIYU
214   CALL MODIFY(IRAD2)
      CALL DRAWI(IXF,IYF-500)
C
C
C
C
      CALL MODIFY(IDUMF)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
      CALL MODIFY(IRAFF1)
      CALL MOVEI(IXRDD,IYYD-300)
      IF(IXEF.GE.0)GO TO 440
      IF(IXEF.LT.0)GO TO 450
440   IXELF=AMOD((IXEF+2000.*SCD),4000.*SCD)-2000.*SCD
      GO TO 460
450   IXELF=AMOD((IXEF-2000.*SCD),-4000.*SCD)+2000.*SCD
460   IGS=IXELF*SCD+CDIS
      IG=IYEF*SCD-1500
      IF (TUNAF.EQ.10000)GO TO 2120
      IF (IGS.GT.2000)GO TO 2120
      GO TO 2110
2120  IGS=IXRDD
      IG=IYYD
2110  CALL MODIFY(IRAFF2)
      CALL DRAWI(IGS,IG-300)
C
C
      IIYU=YU*VAL+100.
      IIXU=0.
      CALL MODIFY(IDUMF1)
      CALL LDTRN3(1.,0.,0.,0.,0.,1.,0.,0.)
      CALL MODIFY(IRAF1)
      CALL MOVEI(IIXU,IIYU-500)
      DIMF=ABS((XEF-XX)/SCALEU)
      IF(DIMF.GT.30000.)GO TO 2130
      IXF=(XEF-XX)/SCALEU
      IYF=(YEF)/SCALEU+100-YU*(1-VAL)
      IF (TUNAF.EQ.10000)GO TO 2130
      IF(IXF.GT.2000)GO TO 2130
      GO TO 2140
2130  IXF=IIXU
      IYF=IIYU
2140  CALL MODIFY(IRAF2)
      CALL DRAWI(IXF,IYF-500)
C
      RETURN
      END
```

```
        COMMON/NAME2/XX,YY,ALEU,ALED,CDIS,VAL,INI,UHN,UVN,XE,YE,TUNA
     *  ,XEF,YEF,TUNAF
        COMMON/NAME3/X(100),Y(100),SHIB(100),N,N1
        COMMON/DMABUF/IDUM(3894),TETAB,TUNAB,TETAFB,TUNAFB,HT,VT,ICOM,
     1  TNGT
        DIMENSION ICHAN(2)
        FLAG=0
        FLAGF=0
        INI=0
        TYPE *,'ENTER ICH0,ICH1,NNN,XSTART,LENGTH'
        ACCEPT *,ICH0,ICH1,NNN,XSTART,LENGTH
        TYPE *,'I0MAX,I0MIN'
        ACCEPT *,I0MAX,I0MIN
        TYPE *,'I1MAX,I1MIN'
        ACCEPT *,I1MAX,I1MIN
        OPEN(UNIT=1,NAME='ANFO.DAT',TYPE='OLD')
        READ(1,*)ALEU,ALED
        READ(1,*)CDIS
        READ(1,*)VAL
        READ(1,*)AMAX
        READ(1,*)TETAM
        READ(1,*)TETAC
        READ(1,*)SWEPT
        READ(1,*)TIM,IW
        READ(1,*)NUMER
        READ(1,*)ANGCT
        CLOSE(UNIT=1,DISPOSE='SAVE')
        OPEN(UNIT=2,NAME='OFF.DAT',TYPE='OLD')
        TYPE *,'X,Y'
        ACCEPT *,XINIT,YINIT
        XINIT=XINIT
        YINIT=YINIT
        XMAX=XSTART+LENGTH
C       T=SAMPLING TIME
        T=.1
        AMASS=1075.
        DRAGH=45.5
        TIMCTH=AMASS/DRAGH
        AKH=80./(60.*DRAGH)
C
C
C
C
        A11H=1
        A12H=T-(T**2)/(2*TIMCTH)
        A21H=0
        A22H=1-(T/TIMCTH)+((T/TIMCTH)**2)/2.
C
C
C
C
C
        SIG1H=(.5*(T**2)-(T**3)/(6*TIMCTH))*(AKH/TIMCTH)
        SIG2H=(T-(T**2)/(2*TIMCTH)+((T/TIMCTH)**2)*T/6.)*AKH/TIMCTH


        T=.1
        AMASS=1075.
        DRAGV=86.
        TIMCTV=AMASS/DRAGV
        AKV=130./(60.*DRAGV)
C
C
C
        A11V=1
        A12V=T-(T**2)/(2*TIMCTV)
        A21V=0
        A22V=1-(T/TIMCTV)+((T/TIMCTV)**2)/2.
C
C
```

```
C
      SIG1V=(.5*(T**2)-(T**3)/(6*TIMCTV))*(AKV/TIMCTV)
      SIG2V=(T-(T**2)/(2*TIMCTV)+((T/TIMCTV)**2)*T/6.)*AKV/TIMCTV
      X1=XINIT
      X2=0
      Y1=YINIT
      Y2=0
      UHN=0
      UVN=0
      AN=0.0
      IW=86
      NN=1
      CALL MARK(1,IW,0,IDS)
1     IF(TIM.EQ.0)GO TO 333
      CALL WAIT(IW,0,IDS)
      CALL MARK(1,IW,0,IDS)
C
C
333   IF(ICOM.EQ.1)GO TO 1003
      CALL AIN(ICH0,II0)
      CALL AIN(ICH1,II1)
      IUV=II0/NNN
      IUH=II1/NNN
      IF(IUH.GT.I1MAX) GO TO 3
      IF(IUH.LT.I1MIN) GO TO 4
      UHN=0
      GO TO 2
3     UHN=60
      GO TO 2
4     UHN=-60
C
C
C
C
2     IF(IUV.GT.I0MAX) GO TO 31
      IF(IUV.LT.I0MIN) GO TO 41
      UVN=0
      GO TO 21
31    UVN=+60
      GO TO 21
41    UVN=-60
      GO TO 21
1003  UVN=VT
      UHN=HT
21    CALL MOTOR7(NUMER,TETAD,SWEPT,TETAM,TETAC,ANGCT)
      TETA=TETAD*3.1415/180.
C
C
      X1N=A11H*X1+A12H*X2+SIG1H*UHN
      X2N=A21H*X1+A22H*X2+SIG2H*UHN
      X1=X1N
      X2=X2N
      XX=X1
C
C
C
C
      Y1N=A11V*Y1+A12V*Y2+SIG1V*UVN
      Y2N=A21V*Y1+A22V*Y2+SIG2V*UVN
      Y1=Y1N
      Y2=Y2N
      YY=Y1
      IF(XX.LT.XSTART)GO TO 122
      WRITE(2,*)XX,YY
122   CALL SONM7(XX,YY,TETAD,TUNA,XE,YE,FLAG,AMAX)
      CALL SONF7(XX,YY,TETAFB,TUNAF,XEF,YEF,FLAGF,AMAX)
      FLAG=1
      FLAGF=1
      CALL SUBM8
      IF(XX.LT.XMAX)GO TO 1
      CLOSE(UNIT=2,DISPOSE='SAVE')
      SUBROUTINE SONM7(XX,YY,TETAD,TUNA,XE,YE,FLAG,AMAX)
      COMMON/DMABUF/IDUM(3894),TETAB,TUNAB,TETAFB,TUNAFB,HT,VT,ICOM,
     1 TNGT
      DIMENSION NUM(100)
      COMMON/NAME3/X(100),Y(100),SHIB(100),N,N1
      IF (FLAG.EQ.1)GO TO 123
      OPEN(UNIT=1,NAME='POINT.DAT',TYPE='OLD')
      READ(1,*)N
      N1=N-1
C     ---------------------------------
C     READ THE POINTS
C
      DO 1 I=1,N
1     READ(1,*)NUM(I),X(I),Y(I)
C
      CLOSE(UNIT=1,DISPOSE='SAVE')
C     ------------------------------------
C
C     REVERSE THE DATA JUST FOR CONVENIENCE
C
      NIM=N/2
      I=1
      NG=N
3     SUBX=X(I)
      SUBY=Y(I)
      X(I)=X(NG)
      X(NG)=SUBX
      Y(I)=Y(NG)
      Y(NG)=SUBY
      I=I+1
      NG=NG-1
      IF(NG.NE.NIM)GO TO 3
      DO 155 I=1,20
155   TYPE *,NUM(I),X(I),Y(I)
C
C     ----------------------
C     CALCULATE THE  TANGENTS
C     THE TANGENTS ARE JUST USEFUL TO FIND THE INTERSECTION POINTS
C     BETWEEN THE LINE SEGMENTS OF THE ENVIRONMENT AND THE  BEAM.
C
      DO 2 I= 1,N1
      I1=I+1
2     SHIB(I)=(Y(I1)-Y(I))/(X(I1)-X(I))
C
C     --------------------------------
C
C     FETCH THE VALUES FOR THE LOCATION OF THE VEHICLE.
C
C
C
C     --------------------------------
C
C     I ASSUME THE VALUE THAT THE BEAM GETS IS EQUAL 5000.
C
      TYPE *,'I DID THE CALCULATION'
C
C
123   IF(XX.GE.0)GO TO 980
      IF(XX.LT.0)GO TO 981
C
C
980   XRD=AMOD((XX+2000.),4000.)-2000.
      GO TO 982
981   XRD=AMOD((XX-2000.),-4000.)+2000.
982   XX=XRD
      TETA=TETAD*3.14159/180
      EXACT=AMAX
C
C     FIND THE INTERSECTION BETWEEN THE BEAM AND EVERY LINE.
C     I=1 MEANS LINE #1
C     I=2 MEANS LINE #2
```

```
        I=0
6       I=I+1
        I1=I+1
C
C       WHEN I=N , THE INTERSECTION BETWEEN THE BEAM AND THE
C       LAST SEGMENT OF THE ENVIRONMENT HAS BEEN CALCULATED.
C       AND THE PROGRAM SHOULD GO TO 13
C
        IF (I.EQ.N)GO TO 13
C
C       ------------------------------------
C       CALCULATE THE VALUE FOR THE INTERSECTION POINT.
C
        IF (ABS(TETAD).EQ.90)GO TO 8
        A=TAN(TETA)*XX-YY
        B=-SHIB(I)*X(I)+Y(I)
        C=TAN(TETA)-SHIB(I)
        IF(ABS(C).GT.0.0001)GO TO 120
        EX1=EXACT*ABS(A+B)/(A+B)
        EX2=-EX1
        IF (C.GT.0.AND.C.LT..0001)XIN=EX1
        IF (C.LT.0.AND.C.GT.-.0001)XIN=EX2
        GO TO 121
120     XIN=(A+B)/C
C       ------------------------------------
C
C       FOR EVERY LINE CHECK IF THE INTERSECTION POINT IS IN FRONT OF
C       THE VEHICLE OR NOT.
C
121     IF (XIN.LE.XX) GO TO 13
C
C       IF NOT IGNORE THAT POINT AND GO TO 13
C
        GO TO 9
8       XIN=XX
C
C       ----------------------------------------
C
C       CHECK IF THE POINT LIES ON THE LINE OR NOT.
C
9       G=(XIN-X(I1))*(XIN-X(I))
        IF(G.LE.0)GO TO 5
        GO TO 13
5       YIN=SHIB(I)*(XIN-X(I))+Y(I)
C
C       DIS=DISTANCE BETWEEN THE VEHICLE AND THE ENVIRONMENT
C
        DIS=SQRT((XX-XIN)**2+(YY-YIN)**2)
        IF (DIS.GT.EXACT)GO TO 13
        EXACT=DIS
        XE=XIN
        YE=YIN
13      IF(X(I).GT.-2000)GO TO 6
        IF(EXACT.NE.AMAX)GO TO 14
        TUNA=10000
        XE=0
        YE=0
        GO TO 7
14      TUNA=EXACT
7       TUNAB=TUNA
        TETAB=TETAD
        RETURN
        END
```

```
      SUBROUTINE MOTOR7(NUMER,TETAD,SWEPT,TETAM,TETAC,ANGCT)
      COMMON/DMABUF/IDUM(3894),TETAB,TUNAB,TETAFB,TUNAFB,HT,VT,ICOM,
     1 TNGT
      IF (NUMER.EQ.1000)GO TO 33
      IS=SWEPT*10.
      IF(NUMER.EQ.IS)NUMER=0
      TETAD=TETAM*COS(2.*3.1415*NUMER/IS)/2.-TETAC
      NUMER=NUMER+1
      GO TO 34
33    TETAD=ANGCT
34    RETURN
      END
      COMMON/DMABUF/IDUM(3894),TETAB,TUNAB,TETAFB,TUNAFB,HT,VT,ICOM,
     1 TNGT
      REAL YOU(10)
C     TETAB=ANGLE OF THE MOVING RADAR
C     TUNAB=THE VALUE OF THE MOVING RADAR
C     TETAFB=ANGLE OF THE FIXED RADAR
C     TUNAFB=THE VALUE OF THE FIXED RADAR
C     HT=VOLTAGE TO TEH HORIZENTAL MOTOR
C     VT=VOLTAGE TO THE VERTICAL MOTOR
C     TNGT=TANGENT OF THE ENVIRONMENT
C
C     THIS SOFTWARE ANALYZES THE DATA COMING FROM THE MOVING RADAR
C     AND ASSIGNS ANGLE FOR THE FIXED RADAR.
C
C
C     I=1   ; CLOCK WORKS
C     I=0   ; CLOCK DOES NOT WORK
      I=1
      IW=6
      TETAS=-89
      STACK=0
      DO 67 I=1,10
67    YOU(I)=0
      TUNAFB=0
C     WHEN THE MOVING RADAR REACHES TO TETAS ONE LOOP OF THE ANALYSIS IS
C     DONE AND THE COMMAND FOR THE FIXED RADAR WILL BE SENT.
C
C
      TYPE *,'ENTER ALP,CTE  40 AND 20 ARE GOOD CHOICES'
      ACCEPT *,ALP,CTE
      TYPE *,'EMERGENCY ANGLE,QWER,TA'
      ACCEPT *,EMERG,QWER,TA
C
C     CTE=VERTICAL DISTANCE TO THE ENVIRONMENT
C     ALP=RADAR DISTANCE TO THE ENVIRONMENT
C
C
C
C
C     CALCULATE THE TANGENT OF THE ENVIRONMENT
C
C
C
2     SIGXY=0.
      SIGX=0.
      SIGY=0.
      SIGX2=0.
      N=0
1     IF(I.EQ.0)GO TO 3
      CALL WAIT(IW,0,IDS)
      CALL MARK(1,IW,0,IDS)
C
C     ANY VALUE OF THE MOVING RADAR WHICH IS TOO LARG CAN BE CONSIDERED.
C     THEREFORE IF THE VALUE THAT COMES FROM THE MOVING RADAR IS 10000
C
C     GO TO 1
3     IF(TUNAB.EQ.10000)GO TO 1
C
C
C     THE ANGLE OF THE MOVING RADAR 'TETAB'IS GRABED. 'TETAF' IS EQUAL THE
C     TETAB
C
      TETAF=TETAB
      ANG=TETAF*3.1415/180.
C
C
C     THE VALUE OF THE MOVING RADAR 'TUNAB' IS FETCHED. 'AMP' IS EQUAL
C     TO 'TUNAB'
C
      AMP=TUNAB
C
```

```
C       CALCULATION FOR TANGENT OF THE ENVIRONMENT
C
        X=AMP*COS(ANG)
        Y=AMP*SIN(ANG)
        SIGXY=SIGXY+X*Y
        SIGX=SIGX+X
        SIGY=SIGY+Y
        SIGX2=SIGX2+X**2
C
        N=N+1
C
C
C       WHEN THE ANGLE OF THE MOVING RADAR IS EQUAL TO TETAS THEN SEND
C       TANGENT OF THE ENVIRONMENT TO CALCULATE TEH ANGLE OF THE FIXED RADAR
C
C
        IF(TETAF.NE.TETAS)GO TO 1
        ANUM=SIGXY-SIGX*SIGY/N
        ADNUM=SIGX2-(SIGX**2)/N
        IF(ABS(ADNUM).LT..001)GO TO 2
C
C
C       TNGT= TANGENT OF THE ENVIRONMENT
C
C
        TNGT=ANUM/ADNUM
        ER=-(.1/TA)
        A=EXP(ER)
        B=1-EXP(ER)
C
C       SEND THE TANGENT OF THE ENVIRONMENT TO CALCULATE THE ANGLE OF THE
C       FIXED RADAR.
C
        CALL MATH(TNGT,ALP,CTE,SHIB)
        OPTA=ATAN(SHIB)*360./3.1415
        IF (TUNAFB.NE.10000)GO TO 345
        YN=EMERG
        GO TO 346
345     YN=YB*A+B*OPTA
        IF (YN.GT.YB)GO TO 346
        DY=ABS(YN-YB)
        YN=YN-DY*QWER
        OPTB=OPTA
346     YB=YN
        TETAFB=YN
C       SHIB=OPTIMUM ANGLE OF THE FIXED RADAR
C
C
        GO TO 2
        END
```

```
      REAL MOM,KA,K1,K2,K3,M1,N1,P1,M2,N2,P2,DN,NU,N
      REAL LAST1,LAST2,LAST3,L1,L2,L3
      DIMENSION IIR(3),AIIR(3),ICHAN(5),AIR(5)
      COMMON/NAME/U7,U8,U9,U7B,U8B,U9B,AMX,AMXB,AMY,
     1 AMYB,AMZ,AMZB,T,DISTY,DISTZ,TX,TY,TZ,ANG,EL,ELB,X11B
      COMMON/DMABUF/IDUM(3894),II(7),I1(7),I2(7),I3(7),IS(7),
     1 F1,F2,ICTL,X(18),ALNGT,AMP
C
C     TAU=TIME CONSTANT OF THE THRUSTERS
C     DAMP=LINEARIZED DAMPING COEFF.
C     GAIN1=DC GAIN OF THE VEHICLE .
C     GAIN2=DC GAIN OF THE THRUSTERS.
C
      TYPE *,'ENTER TAU IN SECONDS.'
      ACCEPT *,TAU
      TYPE *,'ENTER W1,ZITA1,PR1'
      ACCEPT *,W1,ZITA1,PR1
      TYPE *,'ENTER SAMPLING TIME IN MILISECONDS.    SAMPLING TIME SHOULD
     1  NOT BE SMALLER THAN 30 MILISECONDS.'
      ACCEPT *,TT
      TYPE *,'ENTER THE LENGTH OF THE DIST,AMP'
      ACCEPT *,ALNGT,AMP
      TYPE *,'ENTER THE XMIN,XMAX'
      ACCEPT *,XMIN,XMAX
      TYPE *,'ENTER YMIN,YMAX'
      ACCEPT *,YMIN,YMAX
      TYPE *,'ENTER THE DISTANCE FROM THE BOTTOM'
      ACCEPT *,DISTA
      T=TT/1000.
      AMX=0.0
      AMY=0.0
      AMZ=0.0
      AMXB=0.0
      AMYB=0.0
      AMZB=0.0
      U7B=0.0
      U8B=0.0
      U9B=0.0
      U7=0.0
      U8=0.0
      U9=0.0
C
C     M1,N1,P1 ARE THE COEFFICIENTS OF THE CLOSED LOOP CHARACTERISTIC
C     EQUATION.
C     M1=COEFFICIENT OF   [Z**2]
C     N1=COEFFICIENT OF   [Z**1]
C     P1=COEFFICIENT OF   [Z**0]
C     WD1=DAMPED NATURAL FREQUENCY
C
      WD1=W1*SQRT(1-ZITA1**2)
      PPR1=EXP(-PR1*T)
      ED1=EXP(-ZITA1*W1*T)
      M1=-2*ED1*COS(WD1*T)-PPR1
      N1=ED1**2+2.*ED1*PPR1*COS(WD1*T)
      P1=-PPR1*(ED1**2)
      TYPE *,'NATURAL FREQUENCY    W1=',W1
      TYPE *,'DAMPING OF THE SYSTEM      ZITA1=',ZITA1
      TYPE *,'THIRD POLE      PR1=',PR1
      TYPE *,'M1=',M1
      TYPE *,'N1=',N1
      TYPE *,'P1=',P1
C
C     M2 ,N2,P2 ARE THE COEFFICIENTS OF THE OBSERVER CHARACTERISTIC
C     EQUATION.
C     M2=COEFFICIENT OF   [Z**2]
C     N2=COEFFICIENT OF   [Z**1]
C     P2=COEFFICIENT OF   [Z**0]
      ZITA2=ZITA1
      W2=5.*W1
      PR2=5.*PR1
      WD2=W2*SQRT(1-ZITA2**2)
      PPR2=EXP(-PR2*T)
      ED2=EXP(-ZITA2*W2*T)
      TYPE *,'WD2=',WD2,'PPR2=',PPR2,'ED2=',ED2
      M2=-2*ED2*COS(WD2*T)-PPR2
      TYPE *,'ED2=',ED2,'PPR2',PPR2,'WD2=',WD2
      N2=ED2**2+2.*ED2*PPR2*COS(WD2*T)
      P2=-PPR2*(ED2**2)
      TYPE *,'W2=',W2,'ZITA2=',ZITA2,'PR2=',PR2,'M2=',M2,'N2=',N2,'P2=',P2
      DAMP=24.
      GAIN1=1.
      GAIN2=1.
      GAIN=GAIN1*GAIN2
      DO 333 I=1,18
333   X(I)=0.0
      MOM=4.
      T1=TAU
      T2=MOM/DAMP
      TYPE *,'T1=',T1,'T2=',T2
      KA=GAIN/DAMP
      TYPE *,'KA=',KA,'R1=',R1,'R2=',R2,'R3=',R3,'R4=',R4
      GA=EXP(-T/T1)
      GB=EXP(-T/T2)
      TYPE *,'GA=',GA,'GB=',GB
      B3=0.
      B2=0.
      B1=(1-GA)*KA*(T-T2*(1-GB))
      B0=(1-GA)*KA*(-T*GB+T2*(1-GB))
      A2=-GA-GB-1.
      A1=GA*GB+GA+GB
      A0=-GA*GB
      K1=-A2+M1
      K2=-A1+N1
      K3=-A0+P1
      TYPE *,'B3=',B3,'B2=',B2,'B1=',B1,'B0=',B0
      TYPE *,'A2=',A2,'A1=',A1,'A0=',A0
      TYPE *,'K1=',K1,'K2=',K2,'K3=',K3
      H1=B2-B3*A2
      H2=B1-B3*A1
      H3=B0-A0*B3
      TYPE *,'H1=',H1,'H2=',H2,'H3=',H3
      C1=H1
      C2=-H1*A2+H2
      C3=H1*(A2**2-A1)-A2*H2+H3
      C4=H2
      C5=-A1*H1+H3
      C6=H1*(A2*A1-A0)-A1*H2
      C7=H3
      C8=-A0*H1
      C9=A2*A0*H1-A0*H2
      DET=C1*(C5*C9-C6*C8)-C4*(C2*C9-C3*C8)+C7*(C2*C6-C3*C5)
      TYPE *,'C1=',C1,'C2=',C2,'C3=',C3,'C4=',C4,'C5=',C5
      TYPE *,'C6=',C6,'C7=',C7,'C8=',C8,'C9=',C9,'DET=',DET
      LAST1=(C4*C8-C7*C5)/DET
      LAST2=(C2*C7-C1*C8)/DET
      LAST3=(C1*C5-C2*C4)/DET
      TYPE *,'LAST1=',LAST1,'LAST2=',LAST2,'LAST3=',LAST3
      CHAR1=-A2*(A2**2-A1)+A2*A1-A0+M2*(A2**2-A1)-N2*A2+P2
      CHAR2=A2**2-A1-M2*A2+N2
      CHAR3=-A2+M2
      CHAR4=-A1*(A2**2-A1)+A0*A2+M2*(A2*A1-A0)-N2*A1
      CHAR5=(A2*A1-A0)-M2*A1+P2
      CHAR6=-A1+N2
      CHAR7=-A0*(A2**2-A1)+M2*A0*A2-N2*A0
      CHAR8=A2*A0-M2*A0
      CHAR9=-A0+P2
      L1=LAST1*CHAR1+LAST2*CHAR4+LAST3*CHAR7
      L2=LAST1*CHAR2+LAST2*CHAR5+LAST3*CHAR8
      L3=LAST1*CHAR3+LAST2*CHAR6+LAST3*CHAR9
      TYPE *,'L1=',L1,'L2=',L2,'L3=',L3
      C1=1+A2+K1+L1*H1
      C2=L2*H1-1
      C3=L3*H1
      C4=A1+K2+L1*H2
      C5=1+L2*H2
      C6=L3*H2-1
      C7=A0+K3+L1*H3
      C8=L2*H3
      C9=L3*H3+1
      DET=C1*(C5*C9-C6*C8)-C4*(C2*C9-C3*C8)+C7*(C2*C6-C3*C5)
      TYPE *,'C1=',C1,'C2=',C2,'C3=',C3,'C4=',C4,'C5=',C5
      TYPE *,'C6=',C6,'C7=',C7,'C8=',C9,'C9=',C9,'DET=',DET
      D1=(C5*C9-C6*C8)/DET
      D2=(C3*C8-C2*C9)/DET
      D3=(C2*C6-C3*C5)/DET
      D4=(C6*C7-C4*C9)/DET
      D5=(C1*C9-C3*C7)/DET
      D6=(C3*C4-C1*C6)/DET
      D7=(C4*C8-C5*C7)/DET
      D8=(C2*C7-C1*C8)/DET
      D9=(C1*C5-C2*C4)/DET
      TYPE *,'D1=',D1,'D2=',D2,'D3=',D3,'D4=',D4,'D5=',D5,'D6=',D6
      TYPE *,'D7=',D7,'D8=',D8,'D9=',D9,'DET=',DET
      NU=K1*(L1*D1+L2*D4+L3*D7)+K2*(L1*D2+L2*D5+L3*D8)+K3*(L1*D3+L2*D6
      1  +L3*D9)+B3
      DN=-K1*D1-K2*D2-K3*D3+1
      N=NU/DN
      TYPE *,'NU=',NU,'DN=',DN
      TYPE *,'N=',N
      DO 35 I=1,18
35    X(I)=0.
      X71=0
      X72=0
      X73=0
      X81=0
      X82=0
      X83=0
      X91=0
      X92=0
      X93=0
      U7B=0
      U8B=0
      U9B=0
      AMXB=0
      AMYB=0
      AMZB=0
      SHIB=(YMAX-YMIN)/(XMAX-XMIN)
      IM=10
      CALL ANINIT
1     CALL WAIT(IM,1,IDS)
      A10=SECNDS(0)
      CALL AINSQ(2,6,ICHAN)
      CALL AIN(10,IBOOM)
      DO 816 I=1,5
      AIR(I)=SHIB*(ICHAN(I)-XMAX)+YMAX
816   IF(ABS(AIR(I)).LT.0.2) AIR(I)=0
      TZ=AIR(1)
      TX=AIR(2)
      TY=AIR(3)
      CALL COMNDS
      COMND7=0
      COMND8=ANG
      COMND9=1.570796327
      DISTY=AIR(4)
      DISTZ=AIR(5)
      R7=COMND7
      R8=COMND8
      R9=COMND9
      U7=-K1*X71-K2*X72-K3*X73+N*R7
      U8=-K1*X81-K2*X82-K3*X83+N*R8
      U9=-K1*X91-K2*X92-K3*X93+N*R9
      IF(IBOOM.GT.15000)GO TO 1241
      DO 23 I=1,18
23    X(I)=0
      X(12)=DISTA
      X71=0
      X72=0
      X73=0
      X81=0
      X82=0
      X83=0
      X91=0
      X92=0
      X93=0
      R7=0
      R8=0
      R9=0
      U7=0
      U8=0
      U9=0
1241  CALL MDL6
      Z71=-(A2+L1*H1+K1)*X71-(A1+L1*H2+K2)*X72-(A0+L1*H3+K3)*X73+L1*X(7)+N*R7
      Z72=(1-L2*H1)*X71-L2*H2*X72-L2*H3*X73+L2*X(7)
      Z73=-L3*H1*X71+(1-L3*H2)*X72-L3*H3*X73+L3*X(7)
      X71=Z71
      X72=Z72
      X73=Z73
      Z81=-(A2+L1*H1+K1)*X81-(A1+L1*H2+K2)*X82-(A0+L1*H3+K3)*X83+L1*X(8)+N*R8
      Z82=(1-L2*H1)*X81-L2*H2*X82-L2*H3*X83+L2*X(8)
      Z83=-L3*H1*X81+(1-L3*H2)*X82-L3*H3*X83+L3*X(8)
      X81=Z81
      X82=Z82
      X83=Z83
      Z91=-(A2+L1*H1+K1)*X91-(A1+L1*H2+K2)*X92-(A0+L1*H3+K3)*X93+L1*X(9)+N*R9
      Z92=(1-L2*H1)*X91-L2*H2*X92-L2*H3*X93+L2*X(9)
      Z93=-L3*H1*X91+(1-L3*H2)*X92-L3*H3*X93+L3*X(9)
      X91=Z91
      X92=Z92
      X93=Z93
      GG1=SECNDS(A10)
      GG1=GG1*1000
      IM=TT-GG1
      IF(IM.LT.0) GO TO 99
      GO TO 1
99    TYPE *,'SMALL SAMPLE TIME'
      END
      SUBROUTINE MDL6
      COMMON /NAME/U7,U8,U9,U7B,U8B,U9B,AMX,AMXB,AMY,
     1 AMYB,AMZ,AMZB,T,DISTY,DISTZ,TX,TY,TZ,ANG,EL,ELB,X11B
      COMMON/DMABUF/IDUM(3894),II(7),I1(7),I2(7),I3(7),IS(7),
     1 F1,F2,ICTL,X(18),ALNGT,AMP
      REAL A(12),RN1(12),RN2(12),RN3(12),Y(12)
      REAL K1,K2,K3,M1,M2,M3,L1,L2,L3
      TAU=2.
      PO=EXP(-T/TAU)
      AMX=PO*AMXB+(1-PO)*U7B
      AMY=PO*AMYB+(1-PO)*U8B
      AMZ=PO*AMZB+(1-PO)*U9B
      AMXB=AMX
      AMYB=AMY
      AMZB=AMZ
      U7B=U7
      U8B=U8
      U9B=U9
      AIX=4.0
      AIY=4.0
      AIZ=4.0
      H=T
      WGT=6.
      DRX=3.14
      DRY=3.14
      DRZ=3.14
      DRXX=24.0
      DRYY=24.0
      DRZZ=24.0
35    J=0
      DO 41 I=1,12
41    Y(I)=X(I)
18    J=J+1
      DO 56 I=7,9
56    IF(ABS(Y(I)).GE.6.28318531) Y(I)=0
      TV=SQRT(Y(1)**2+Y(2)**2+Y(3)**2)
      A(1)=(TX-DRX*Y(1)*TV)/WGT-Y(3)*Y(5)+Y(2)*Y(6)
      A(2)=(TY-DRY*Y(2)*TV)/WGT-Y(1)*Y(6)+Y(3)*Y(4)
      A(3)=(TZ-DRZ*Y(3)*TV)/WGT-Y(2)*Y(4)+Y(1)*Y(5)
      A(4)=(AMX-DRXX*Y(4)-(AIZ-AIY)*Y(5)*Y(6))/AIX
      A(5)=(AMY-DRYY*Y(5)-(AIX-AIZ)*Y(4)*Y(6))/AIY
      A(6)=(AMZ-DRZZ*Y(6)-(AIY-AIX)*Y(4)*Y(5))/AIZ
      A(7)=Y(4)+(Y(5)*SIN(Y(7))+Y(6)*COS(Y(7)))*TAN(Y(8))
      A(8)=Y(5)*COS(Y(7))-SIN(Y(7))*Y(6)
      A(9)=(Y(5)*SIN(Y(7))+Y(6)*COS(Y(7)))/COS(Y(8))
      A(10)=COS(Y(9))*COS(Y(8))*Y(1)+
     1 (COS(Y(9))*SIN(Y(8))*SIN(Y(7))-SIN(Y(9))*COS(Y(7)))*Y(2)+
     2 (COS(Y(9))*SIN(Y(8))*COS(Y(7))+SIN(Y(9))*SIN(Y(7)))*Y(3)
      A(11)=SIN(Y(9))*COS(Y(8))*Y(1)+
     1 (SIN(Y(9))*SIN(Y(8))*SIN(Y(7))+COS(Y(9))*COS(Y(7)))*Y(2)+
     2 (SIN(Y(9))*SIN(Y(8))*COS(Y(7))-COS(Y(9))*SIN(Y(7)))*Y(3)
      A(12)=-SIN(Y(8))*Y(1)+COS(Y(8))*SIN(Y(7))*Y(2)+
     1 COS(Y(8))*COS(Y(7))*Y(3)
      GO TO (50,60,70,80)J
50    DO 55 I=1,12
      RN1(I)=A(I)
55    Y(I)=X(I)+H*RN1(I)/2.
      GO TO 18
60    DO 65 I=1,12
      RN2(I)=A(I)
65    Y(I)=X(I)+H*RN2(I)/2.
      GO TO 18
70    DO 75 I=1,12
      RN3(I)=A(I)
75    Y(I)=X(I)+H*RN3(I)
      GO TO 18
80    DO 85 I=1,12
      X(I)=X(I)+H*(RN1(I)+2*RN2(I)+2*RN3(I)+A(I))/6.
85    CONTINUE
      DO 87 I=7,9
      NDIME=X(I)/6.281385
87    X(I)=X(I)-NDIME*6.283185
      AA=SIN(X(7))
      BB=SIN(X(8))
      CC=SIN(X(9))
      DD=COS(X(7))
      EE=COS(X(8))
      FF=COS(X(9))
      K1=EE*FF*X(1)
      K2=(FF*BB*AA-CC*DD)*X(2)
      K3=(FF*BB*DD+CC*AA)*X(3)
      X(13)=K1+K2+K3
      L1=CC*EE*X(1)
      L2=(CC*BB*AA+FF*DD)*X(2)
      L3=(BB*CC*DD-FF*AA)*X(3)
      X(14)=L1+L2+L3
      M1=-BB*X(1)
      M2=EE*AA*X(2)
      M3=EE*DD*X(3)
      X(15)=M1+M2+M3
      X(16)=X(4)+(X(5)*AA+X(6)*DD)*BB/EE
      X(17)=X(5)*DD-AA*X(6)
      X(18)=(X(5)*AA+X(6)*DD)/EE
C     WRITE(5,*)(X(I),I=7,12)
      RETURN
      END
```